UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

Comparación de Algoritmos de Cálculo de Ratios de Verosimilitudes para Interpretación Forense

Juan Maroñas Molano
Tutor: Daniel Ramos Castro

Junio 2015

Comparación de Algoritmo de Cálculo de Ratios de Verosimilitudes para Interpretación Forense

Juan Maroñas Molano
Tutor: Daniel Ramos Castro

Biometric Recognition Group – ATVS
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2015

Resumen

Este TFG compara varios algoritmos de cálculo de ratios de verosimilitudes (LR), en un entorno de casos forenses reales. El marco de aplicación son los sistemas automáticos de reconocimiento de locutores utilizados en casos forenses reales. Los LR son actualmente la recomendación para emitir conclusiones evaluativas en informes forenses en Europa.

En la primera parte del TFG se hace uso de tres métodos populares para calcular LR a partir de puntuaciones de sistemas biométricos (llamadas "scores" en inglés): regresión logística (LogReg), modelado gaussiano de máxima verosimilitud (Gauss-ML) y Kernel Density Function Gaussianas (KDF). La comparación de estos métodos se lleva a cabo en el contexto de la evaluación NIST 2012 de reconocimiento de locutor (NIST SRE 2012), donde se presentan varios escenarios de habla conversacional telefónica y microfónica realista. El análisis pone de manifiesto problemas típicos en este tipo de entornos, como el desajuste de bases de datos, el sobreajuste de los modelos, etc.

En la segunda parte del TFG se propone el uso del modelado gaussiano bayesiano (Gauss-Bayes), propuesto recientemente como alternativa a los tres métodos analizados anteriormente. Se concluye que, cuando existe gran cantidad de datos de entrenamiento de los modelos, este método equivale a Gauss-ML. Sin embargo, si los datos de entrenamiento son escasos, Gauss-Bayes supera ampliamente en rendimiento a Gauss-ML.

Posteriormente, el TFG presenta una aportación original de aplicación: se propone un escenario real en ciencia forense, en el que las proposiciones definidas en el caso dan lugar a una falta considerable de scores de entrenamiento para los modelos. Este caso es muy común en ciencia forense, tal y como se describe. Se muestra que el método Gauss-Bayes supera con creces al método Gauss-ML en este escenario, dando lugar a cálculos de LR robustos y coherentes. Más aún, se proponen dos esquemas de cálculo de scores de entrenamiento (llamados también esquemas de anclaje, o "anchoring") que pueden ser adecuados para el uso en casos forenses reales, y se presenta el rendimiento para ambos esquemas, constatando que Gauss-Bayes es la mejor opción en dichos escenarios. Los resultados de esta sección son de relevancia, y previsiblemente se enviarán para publicación tras la finalización del TFG.

Abstract

This TFG compares several algorithm for likelihood ratio (LR) computation, in real forensic enviroments. The implementation framework is automatic speaker recognition systems used for real forensic cases. LR are actually recommended to issue evaluative conclusions on forensic reports in Europe.

In the first part, three popular LR computation methods are used for calculating LR based on scores from a biometric system: Logit Regression (LogReg), maximum likelihood Gaussian modeling (Gauss-ML) and Gaussian Kernel Density Funtion (KDF). Comparison of these methods is carried out in the context of the 2012 NIST speaker recognition evaluation (NIST SER 2012), where various call scenarios and realistic conversational speech microphone are present. This analysis shows tipical problems in this scenarios like database mismatch, model overfitting...

In the second part it is proposed the use of bayesian gaussian modeling , recently proposed as an alternative for the three methods proposed before. It is concluded that when there is lots of train data this methods is equivalent to Gauss-ML. However, if there is few train data, bayesian method outperforms widely Gauss-ML.

Finally, this TFG shows an original framework application: a real forensic scenario is proposed in which de propositions defined by the case imply a considerable lack of train scores. This is a very tipical case in forensic science, as it is described. It is showed that bayesian method is much better tan Gauss-ML in this kind of scenario, computing robust and coherent LR. Furthermore, two score computing schemes (also known as anchoring schemes) can be adecuated for being used in real forensic cases, and performance it is presented for both schemes, noting that bayesian estimation is the best choice in these scenarios. Results from these section are relevant, and likely will be submitted for publication after the end of the TFG.

Juan Maroñas Molano Universidad Autónoma de Madrid
Daniel Ramos Castro

Agradecimientos

Este trabajo va a dedicado a todo el círculo de personas que han compartido estos 4 bonitos años. Aquí acaba un ciclo y comienza otro. Gracias a todo el mundo.

Palabras Clave

Ratio de verosimilitud

Probabilidad predictiva

Teoría Bayesiana de la decisión

Calibración

Poder Discriminante

Interpretación Forense

Evaluación de evidencias

Modelos Bayesianos

Key Words

Likelihood ratio

Predictive probability

Decision Bayes Theory

Calibration

Discriminant Power

Forensic Interpretation

Evidence assessment

Bayesian models

Índice Contenido

Índice Figuras

Glosario

LR: Likelihood Ratio

LLR: Logaritmic Likelihood Ratio

score-LR: transformación Score-LR

s-LR: transformación Score-LR

ADN: ácido desoxirribonucleico

ML: Maximum Likelihood

MB: Modelado Bayesiano

RL: Regresión Logística

FA: Falsa aceptación

FR: Falso Rechazo

log-ratio: Logaritmic Likelihood Ratio

GG: Función densidad de probabilidad Gaussian-Gamma

Gam: Función densidad de probabilidad Gamma

prior-GG: Función densidad de probabilidad Gaussian-Gamma del parámetro a priori

1. Introducción

1.1. Motivación del proyecto

La cienca forense es la disciplina que estudia el uso de procedimientos científicos en casos judiciales. Muchos de estos casos incluyen a personas a las que se está juzgando por un delito. En uno de estos casos, un tomador de decisiones (juez, jurado) tiene que combinar la información del caso referente a la culpabilidad o inocencia de la persona juzgada, con la llamada *prueba* o *evidencia* científica. Esta prueba se analiza por parte de un *especialista* forense, o *perito* forense, que debe generar un informe forense con el resultado de su evaluación de la prueba, con una interpretación adecuada para que lo entienda el juez, (Martyna, Zadora, Ramos, & Aitken, 2014).

Dentro de esta disciplina, la toma de decisiones es una tarea de alta relevancia y complejidad. Es por ello necesario, por un lado, establecer técnicas rigurosas que permitan apoyar la toma de decisiones con la mayor independencia posible respecto de las condiciones particulares de la evidencia del caso (huella dactilar, ADN, locuciones…), y por otro permitir cierta flexibilidad para que el perito forense incorpore información que permita adaptar una técnica general de interpretación a un tipo de evidencia particular del problema bajo estudio.

En la ciencia forense moderna la interpretación de evidencias está basada en los LR (*likelihood ratio* o ratio de verosimilitud) para apoyar la toma de decisiones, (Ramos, Gonzalez-Rodríguez, Zadora, & Aitken, 2012) (Ramos & González-Rodriguez, Reliable support: Measuring calibration of likelihood ratios, 2013) . Cuando se usan sistemas biométricos para analizar la prueba forense (en casos como, por ejemplo, de reconocimiento de personas por su voz, o por sus huellas dactilares), éstos devuelven puntuaciones (*scores*) que sirven para dar una representación cuantitativa de lo que el sistema biométrico está midiendo (por ejemplo similitud entre huellas dactilares). Por tanto en sistemas biométricos el problema del calculo de LR se resuelve a partir de las denominadas transformaciones *score-LR* (en adelante *s-LR* o *score-LR*). El cálculo de esta transformación está basado en análisis estadístico de las pruebas de un caso (huellas, vidrios, voz, ADN…).

En un entorno forense el cálculo de LR presenta múltiples problemas: escasez de datos (voz de muy corta duración, huellas dactilares muy parciales…), condiciones de adquisición de las pruebas no óptimas (huellas dactilares degradadas, locuciones de baja calidad…), variabilidad de los sujetos (hombre, mujer, enfermedades del tracto vocal…), etc. Ello motiva que, aunque se usen técnicas aparentemente correctas para el cálculo de LR, dichos LR pueden llevar a decisiones incorrectas en tanto en cuanto el método de cálculo de dicho valor no es robusto al problema que se pueda presentar en el caso en cuestión (como puede ser la escasez de los datos), (Swart & Brümmer, 2014).

Por tanto es necesario establecer métodos de cálculo de LR robustos y medidas de rendimiento que permitan comprobar el funcionamiento que presenta el método en el entorno forense real.

1.2. Objetivos y Enfoque

El objetivo general de este trabajo es el de probar una técnica de cálculo de LR reciéntemente propuesta, el modelano bayesiano gaussiano, que presenta más robustez para el cálculo de LR que otros métodos anteriores, en un escenario con escasez de datos; y compararla con técnicas de más amplia utilización. Este escenario de falta de datos para entrenar modelos de LR es muy típico en casos reales forenses. Para ejemplificar el problema, nos enmarcamos en el reconocimiento automático de locutor. El enfoque propuesto ha dado lugar a los siguientes objetivos particulares:

1. Análisis de un problema de cálculo de *s-LR*: En una primera parte el objetivo es, mediante el uso de algoritmos de clasificación de patrones muy comunmente utilizados para el cálculo de ratios de verosimilitud, analizar problemas tipo en el uso de dichos métodos de cálculo de LR. Dicho análisis permitirá exponer un conjunto de conclusiones y observaciones que servirá como punto de partida para el resto del TFG, como la selección del método más adecuado o los problemas más relevantes de los métodos analizados que se podrían resolver con otras propuestas.

2. Asignación bayesiana de probabilidades predictivas: En la segunda parte el objetivo es probar una técnica basada en estimación bayesiana para el cálculo de las probabilidades predictivas que permitan describir las clases que toman parte del problema. En ella se expondrán las ventajas hipotéticas de esta técnica frente a otras anteriormente analizadas, además de modelar matemáticamente el problema para intentar demostrar la veracidad de la hipótesis de partida: el correcto funcionamiento para pocos datos de entrenamiento.

3. Entorno forense y el problema del anclaje (*Anchoring*): este objetivo consiste en aplicar los métodos analizados y propuestos en casos forenses reales, donde cobra importancia el concepto de *anchoring* o anclaje. Este concepto esta relacionado con la definición de las proposiciones en un caso forense. Primero se simulará un entorno forense típico, y finalmente se probarán los métodos anteriores, con la hipótesis de que el método bayesiano solucionará algunos problemas presentes derivados del anclaje en casos forenses.

Por tanto la memoria queda estructurada de la siguiente manera:

- Definiciones y Nomenclatura utilizada
- Revisión del estado del arte.
- Descripción de los métodos de cálculo de s-LR.
- Experimentos y Resultados obtenidos para el punto 1 y punto 3 de los objetivos planteados en este apartado.

2. Definiciones y nomenclatura utilizada

Antes de comenzar es necesario realizar una pequeña descripción de la nomenclatura utilizada en las descripciones matemáticas.

- En mayúscula se representan las matrices y en mínuscula los vectores. Además los vectores de más de una dimensión se representan en negrita. Por otro lado la letra mayúscula también se utiliza para definir conjuntos y variables aleatorias. Cuando se defina un conjunto, éste siempre irá acompañado de {} mientras que la matriz irá en negrita.

- El término clase hace referencia a los posibles resultados de las decisiones en el caso forense. Por ejemplo en un entorno de identificador de locutor una clase modela que las locuciones a comparar pertenecen a la misma persona y la otra que no pertenece a la misma persona.

- Las clases se representan mediante la letra H. En este marco de trabajo Hp será la clase *target* y Hd la clase *non-target*, donde *target* representa locuciones pertenecientes al mismo locutor y *non-target* locuciones pertenecientes a distintos locutores.

- Un *score* o puntuación es el número obtenido por un sistema biométrico a partir de las locuciones comparadas y por lo tanto representativo de las mismas. En un problema de dos clases, un score debería ser mayor cuanto más apoye a una clase, y menor cuanto más apoye a la clase contraria.

- El LR es el ratio de verosimilitud utilizado para apoyar las decisiones. El LR es un score con interpretación probabilística. En un contexto de dos clases (hipótesis) en ciencia forense un LR de valor 4 signfiica que "es 4 veces más probable observar la evidencia forense si la hipótesis 1 es cierta que si lo es la hipótesis 2".

- La letra $\boldsymbol{\theta}$ representa el vector de parámetros del modelo que representa la clase. Por ejemplo para una gaussiana $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{V})$ donde el primer parámetro representa el vector de medias y el segundo la matriz de covarianzas.

- Las puntuaciones se representan mediante la letra s. Además un conjunto de scores observadas se representan como $\mathcal{S} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \boldsymbol{s_3}, \dots, \boldsymbol{s_n}\}$. Por lo tanto s representa la variable aleatoria del *score* generado por el sistema, cuya distribución está parametrizada por $\boldsymbol{\theta}$. S representa un valor particular observado de dicho *score*.

- A la hora de describir probabilidades mediante la $p(s)$ minúscula se describen funciones de densidad y mediante la $P(S \leq s)$ mayúscula se describen probabilidades.

- La frontera (también llamada umbral) de decisión se representa mediante ξ y el coste asociado a cometer un error mediante λ.

- Los subíndices fa y fr hacen referencia a los errores de *false acpetance* y *false rejection* o lo que es lo mismo se refieren a *scores non-target* clasificados como *target* y viceversa, respectivamente.

3. Trabajos previos y Estado del Arte

3.1. Proceso de clasificación forense.

Dentro de la interpretación de evidencias con LR utilizando sistemas automáticos de reconocimiento de locutores, un esquema típico se puede dividir en tres etapas.

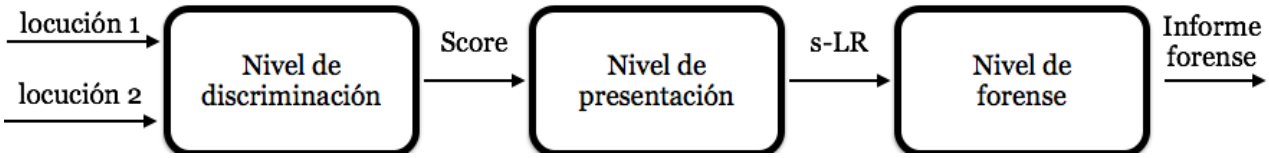


Figura 1: Tres niveles en los que se divide el proceso de interpretación en un caso forense donde se utilizan sistemas de reconocimiento de locutor.

Los elementos básicos de un caso forense de identificador de locutor son: las locuciones y las proposiciones o hipótesis.

Las locuciones se pueden clasificar en dos tipos:

- **Dubitadas:** obtenidas de la escena del crimen y por tanto de carácter incriminatorio. La identidad del sujeto que la genera es el objetivo final a determinar.

- **Indubitadas:** obtenidas a partir del sospechoso del caso, y por tanto de identidad conocida.

Las proposiciones o hipótesis sirven para representar las posibles decisiones que pueden tomar parte en el proceso de identificación. Por ejemplo una proposición en el contexto de reconocimiento de locutor sería que las locuciones a comparar pertenecen a la misma persona.

Además forman parte de estos elementos el sujeto sospechoso así como los sujetos que podrían ser potenciales autores de la toma dubitada, a los que se engloba en la llamada *población de referencia del caso*.

El funcionamiento del sistema se puede resumir de la siguiente manera. A la entrada del sistema se tienen dos locuciones. En un primer paso a partir de la extracción de características de dichas locuciones se calcula una puntuación (o *score*) que tendrá un valor más alto cuanto mayor sea la similitud entre las locuciones. Esta puntuación depende de las locuciones de entrada. Se establecen dos hipótesis: las locuciones pertenecen al mismo locutor (clase Hp o *target*) o a locutores diferentes (clase Hd o *non-target*). Se verá más adelante cómo se generan los datos para modelar las proposiciones del caso en cuestión, en un proceso llamado *anchoring*.

En el siguiente nivel, el nivel de presentación, se generan modelos que permiten describir dichas puntuaciones y por lo tanto describen cada una de las clases del problema. Utilizando los modelos se calcula una transformación *score-LR*.

El nivel forense se refiere a la elaboración del informe forense que comunique la conclusión del perito. Se refiere a de qué forma comunicar, a cómo acreditar procedimientos, a cómo incluir informes de validación de las técnicas, y en definitiva a todo lo necesario para que el informe forense sea aceptado en un juicio. Este nivel no es objetivo de este TFG.

Este TFG se centra en el nivel de presentación, es decir, cómo calcular transformaciones de *s-LR* que resulten adecuadas en casos reales simulados.

3.2. Toma de decisiones bayesiana

El marco de la toma de decisión propuesto en casos forenses en Europa es el marco lógico de decision bayesiana, utilizado en la metodología LR. Es por ello que este apartado se dedica a describirlo, (Duda, Hart, & Stork, Pattern Classification, 2000):

$$\frac{P(Hp|s)}{P(Hd|s)} = \frac{p(s|Hp)}{p(s|Hd)} * \frac{P(Hp)}{P(Hd)} \quad (1)$$

O también puede expresarse como:

$$\log\left(\frac{P(Hp|s)}{P(Hd|s)}\right) = \log\left(\frac{p(s|Hp)}{p(s|Hd)}\right) + \log\left(\frac{P(Hp)}{P(Hd)}\right) \quad (2),$$

En esta expresión el primer miembro relaciona las probabilidades a posteriori de las hipótesis, es decir, la probabilidad de que la clase sea una u otra dependiendo de toda la información presente en el caso forense, incluyendo la evidencia forense. En el segundo miembro se tiene la transformación *score-LR* multiplicada por probabilidades a priori del caso. En las probabilidades a priori se incluye la información dependiente del caso, pero que no incluye la evidencia forense. En la transformación *score-LR* se valora la información correspondiente a la evidencia forense, es decir, se obtiene un *score* del sistema biométrico de reconocimiento de locutor. Se dice que la *s-LR* es independiente de aplicación porque las dos mismas locuciones de las que se obtiene el *score*, podrían formar parte de casos totalmente distintos. Además $p(s|Hp)$ se conoce como la función de verosimilitud o *likelihood*.

La transformación *score-LR* es:

$$\frac{p(s|Hp)}{p(s|Hd)} \quad (3)$$

Cuando la transformación se particulariza a un valor determinado de score observado *S*, se tiene el LR del caso:

$$LR = \frac{p(S|Hp)}{p(S|Hd)} \quad (3b)$$

Es decir, el LR es el ratio de verosimilitudes particularizado en el valor observado del scores S del caso, que es el que se obtiene en la comparación de la toma dubitada con la toma indubitada.

Además debido a que se tienen dos clases complementarias:

$$P(Hp|s) = 1 - P(Hd|s) \quad (4)$$

Por lo tanto introduciendo 4 en 1 se puede obtener la siguiente expresión para la probabilidad a posteriori para una clase:

$$P(Hp|s) = \frac{LR * O(Hp)}{1 + LR * O(Hp)} = \frac{1}{1 + \frac{1}{LR * O(Hp)}} = \frac{1}{1 + e^{-\log(LR*O(Hp))}} \quad (5)$$

Donde $O(Hp) = \frac{P(Hp)}{P(Hd)}$. Nótese que la expresión final es válida cuando el mapeo *score-LR* está expresado en logaritmo natural. Por lo tanto la probabilidad a posteriori se puede representar como una función sigmoide del log(LR). De ahora en adelante las siglas LLR harán referencia al LR expresado de forma logarítmica.

Para tomar deciciones, el tomador de decisiones debe fijar, además, los costes de decisión λ_{fa} y λ_{fr}, que representan el coste asociado a cada decisión incorrecta (falsa aceptación y falso rechazo). Mediante la información a priori y los costes de decisión se establece una frontera de decisión, que en realidad es un umbral escalar debido a que el sistema biométrico empleado en este trabajo arroja puntuaciones 1-dimensional. Si el LR está por encima de dicha frontera se decide una clase y si está por debajo se decide otra. Dicha frontera se denomina *umbral de Bayes*:

$$\xi = \frac{P(Hd)\lambda\text{fa}}{P(Hp)\lambda\text{fr}} \quad (6)$$

Tanto las probabilidades a priori como los costes de decisión son responsabilidad del tomador de decisiones, y el perito forense no debe fijar su valor. Es por ello que a los costes e información a priori se les conoce como parámetros dependientes de aplicación. El perito tiene la responsabilidad de calcular el LR. Todo ello, permite al juez tomar decisiones de forma lógica en el marco bayesiano. Dentro de este marco se puede hablar de mínimo error cometido o mínimo riesgo. El mínimo error es aquel que el marco de toma de decisión bayesiana garantiza que se comete cuando aplica (1). El mínimo riesgo es el que el marco de toma de decisión bayesiana garantiza que se comete cuando se incorpora información acerca de los costes en la toma de decisión, (6). El mínimo riesgo es el mínimo error cometido para unos costes diferentes. Si los costes son iguales el mínimo error y el mínimo riesgo son iguales.

3.3. Transformación *score-LR*: marco forense de cálculo de LR

Dentro de las tres etapas expuestas con anterioridad, éste trabajo está orientado a trabajar en el nivel de presentación. En el marco de decisión bayesiana el objetivo es obtener LR adecuados debido a que es aquí donde se incluye la información proveniente del sistema y por lo tanto es la parte correspondiente al análisis forense de las pruebas. Cualquier otro establecimiento de información sobre el caso es competencia del

tomador de decisiones fijar su relevancia para el caso en cuestión y dicha información está contenida en la información a priori. Cabe señalar que el análisis forense nunca debe dar lugar a una valoración de dichas probabilidades a priori de las clases, y por lo tanto esta valoración es competencia del tomador de decisiones (juez, jurado, etc.).

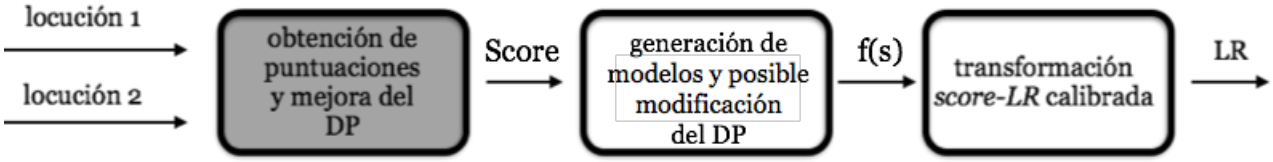


Figura 2: Esta figura muestra un esquema del cálculo de LR utilizando sistemas de reconocimiento de locutores. La caja que obtiene las puntuaciones, de ahora en adelante, será una caja negra. Este trabajo se centra en las dos cajas blancas. Por lo tanto partiremos de un conjunto de scores obtenidas a partir de un sistema de reconocimiento de locutor que calcula scores, y nos centraremos en obtener buenos modelos de cálculo de LR.

Dentro del marco forense de cálculo de LR, el objetivo es transformar un conjunto de *scores* representativos de cualquier problema (identificador de locutores, ADN, huella dactilar…) en un LR interpretable como un grado de apoyo a las proposiciones que toman parte del problema, relacionadas con el hecho de que el sospechoso sea culpable o no.

El LR, además de contener información concerniente a la evidencia o prueba forense, permite apoyar las decisiones con independencia de la naturaleza de las pruebas. Esto quiere decir que ante pruebas y sistemas de extracción de características diferentes el LR permite hacer medidas comparativas entre ambos, porque siempre se interpreta de la misma manera, independientemente de la disciplina forense de la que proceda (ADN, huellas dactilares, voz, etc.).

En el marco en el que este trabajo se desarrolla, esto quiere decir que dado un sistema que obtiene puntuaciones a partir de dos locuciones, con valores más positivos cuánto mayor sea la similitud que establece el nivel de discriminación, el LR permite realizar medidas comparativas de conjuntos con locuciones con distinta naturaleza al representar la información contenida en dichas puntuaciones de la misma manera. Esto es, transforma el conjunto de *scores,* del que no se sabe a partir de qué valor se puede considerar que la locución pertenece a la misma persona o no, en una función que estandariza este apoyo.

El anterior párrafo puede ejemplificarse de la siguiente manera: imagine que se obtienen *scores* en unas condiciones determinadas (por ejemplo voz telefónica a 8KHz en una cafetería), y se fija un umbral de decisión para esas condiciones y esos *scores*. Si cambian las condiciones es posible que cambien los rangos de los *scores* y haga falta un nuevo umbral diferente, que habría que calcular para los nuevos *scores*. Sin embargo la transformación *score-LR* genera un resultado, el LR, que siempre se interpreta de la misma manera, y siempre genera decisiones óptimas para el umbral de Bayes. Por lo tanto, no es necesario volver a ajustar umbrales con datos si se usan LR: simplemente se utilizará el umbral de Bayes, dado por (6) .

Lo comentado en estos últimos tres párrafos tiene que ver con dos propiedades deseables en un conjunto de LR: la calibración y el poder discriminante.

3.4. Calibración y poder discriminante

Dentro de la toma de decisiones Bayesiana son dos los conceptos que influyen en el rendimiento de la generación de la transformación *score-LR*. Ambos conceptos pueden encontrarse en la literatura, (D. van Leeuwen & Brümmer, 2007) (Ramos, Gonzalez-Rodríguez, Zadora, & Aitken, 2012) (Ramos & González-Rodriguez, 2013) (Zadora, Ramos, Martyna, & Aitken, 2014), y se describen a continuación:

- **Poder discriminante:** El poder discriminante (*discriminant power*,DP) es una característica del rendimiento de un conjunto de *scores* o de LRs principalmente relacionado con la etapa de extracción de características y obtención de la puntuación. El DP mide la separación de las distribuciones entre las clases (Figura 3). Cuanto mayor es la separación mayor DP por lo que un buen DP es condición necesaria pero no suficiente para obtener un buen rendimiento. En la etapa de obtención de modelos se puede mejorar o empeorar el DP, es decir, si unos *scores* presentan un determinado DP, una transformación *score-LR* puede alterar ese DP. Por lo tanto, es necesario medir el DP tanto para los *scores* como para los LR.

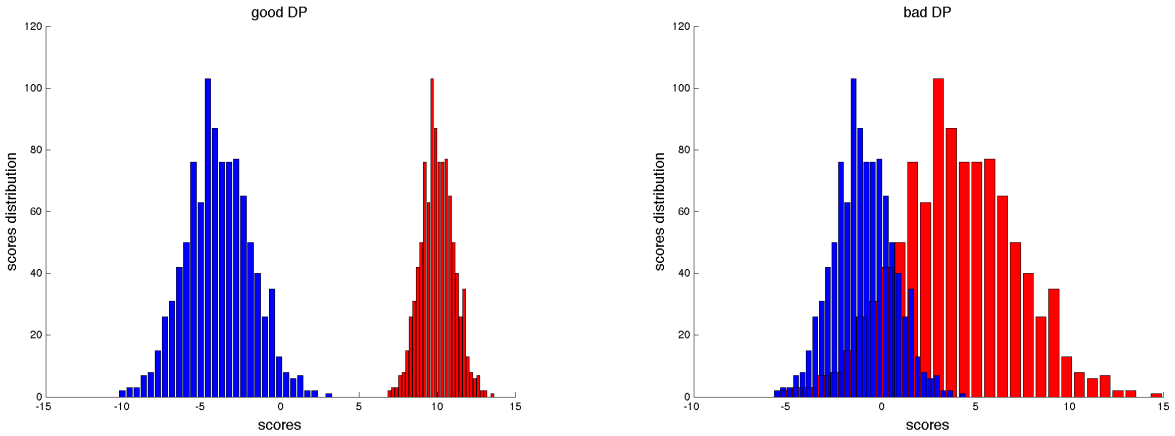


Figura 3: Imágenes de poder discriminante. A la izquierda una imagen con dos poblaciones (scores target en rojo y non-target en azul) con buen DP, a la derecha dos poblaciones con peor DP.

- **Calibración:** La calibración es un concepto que mide el rendimiento del modelo de LR en términos de lo buena o mala que es la interpretación de los LR en un marco de decisión bayesiano. Un conjunto de LR estará bien calibrado cuando, dado una característica observada para la que se obtendrá una probabilidad a posteriori de pertenencia a una clase, la probabilidad de aparición de dicha característica en la población manejada tenderá a ser la probabilidad a posteriori que arroja el clasificador. Un ejemplo de esto sería: dadas 2 clases de una especie marina como las lubinas y los salmones y dada una característica que mide el tamaño a partir de un solo pez, si el sistema está bien calibrado y arroja una probabilidad a posteriori del 25% para la clase lubina dada esa característica entonces el 25% de las muestras de la población que presentan la misma

característica serán lubinas. En otras palabras la calibración mide cómo se ajusta el modelo de LR a la realidad presente en los datos.

Recordando la figura 2, en la aplicación del modelo de cálculo de LR podía haber una modificación del DP de los scores, es decir, el DP de los LR podía ser diferente. Además en el momento que se calcula el LR la transformación pasa a mejorar la calibración.

En este punto cabe destacar que la calibración es un concepto que se puede interpretar de muchas maneras. Se dice también que un *s-LR* está calibrado porque es siempre interpretable de la misma manera. Aun así si el modelado de los datos no es correcto la transformación *s-LR* puede no estar calibrada por lo tanto se hablará de calibración en términos de lo definido en este apartado.

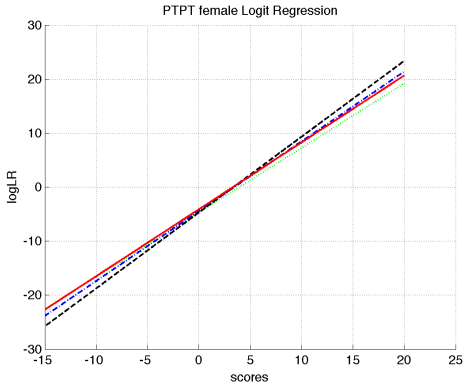


Figura 4: Transformación s-LLR para diferentes conjuntos de datos (scores) de entrenamiento.

Una forma de visualizar gráficamente los modelos de cálculo de LR es mediante la representación de la transformación de *scores* a log(LR), ver Figura 4. En la figura se pueden observar 4 transformaciones *score-LR* obtenidas a partir de cierto modelado (en este caso mediante regresión logística, técnica descrita más adelante). Cada una de estas transformaciones se corresponden con diferentes conjuntos de *scores* de entrenamiento, algunos pertenecientes a la clase Hp y otros a la clase Hd. Cada uno de los conjuntos de scores que generaron cada transformación presentan valores diferentes. Valores de log(LR) (representados en el eje y) por encima de 0 apoyan la pertenencia a la clase Hp y viceversa. Cuanto mayor es el valor mayor es dicho apoyo.

Se observa como el LR permite realizar medidas comparativas entre las 4 transformaciones. El grado de calibración dependerá de los datos y del modelado utilizado manifestado en la variabilidad de las diferentes *s-LR*, tal y como se observa en la figura (debido a que los modelos varían, para un mismo *scores* varía el apoyo a cada una de las hipótesis). Estas cuatro transformaciones han sido obtenidas a partir de locuciones en las mismas condiciones de grabación por lo que cabría esperar *s-LR* iguales y esto no es así, debido a que la transformación depende de las puntuaciones con las que esta se entrena, por lo tanto el apoyo a las decisiones es diferente y por lo tanto el grado de calibración presente también. De hecho, un conjunto de LR bien calibrado presenta una relación entre su DP y la fuerza de dichos LR: cuanto mejor es el DP, más fuertes son los valores de LR (es decir, más alto tiende a ser el |log(LR)|) (Ramos & González-Rodriguez, 2013), o lo que es lo mismo: si un sistema está mal calibrado aunque presente muy buen DP se puede obtener un mal rendimiento del apoyo a la hipótesis,

esto es, transformaciones *s-LR* altas para *scores* que por el valor que toman no deberían tenerlo (cerca de la zona en donde las dos distribuciones se entremezclan, ver Figura 3, gráfica "*BadDP*"). Además el rendimiento del sistema será mejor cuanto mejor sean el DP y la calibración, es decir, el rendimiento global depende de estas dos medidas de rendimiento*,* (D. van Leeuwen & Brümmer, 2007) (Zadora, Ramos, Martyna, & Aitken, 2014) (Ramos, Gonzalez-Rodríguez, Zadora, & Aitken, 2012) (Ramos & González-Rodriguez, 2013).

Dicho esto, en la bondad de la obtención de los LR generados influyen tanto el DP como la calibración. Al generar un modelo de cálculo de LR se puede modificar el DP tanto para bien como para mal dando lugar a *s-LR* que pueden dar lugar a LRs con el mismo o distinto DP, (D. van Leeuwen & Brümmer, 2007). El objetivo es mejorar la calibración sin disminuir el DP que arroja el sistema generador de *scores*.

Una propiedad que se ha estudiado recientemente es que siempre que la transformación *s-LR* sea invertible, se puede asegurar que no se modifica el DP, es decir, que los *scores* del sistema de reconocimiento y los LR generados de dichos *scores* presentan el mismo DP, (D. van Leeuwen & Brümmer, 2007). Además aplicaciones monótonas sobre un espacio vectorial de tipo $T: \Re \rightarrow \Re$ son también invertibles. Por lo tanto dado que las *scores* que arroja el sistema son unidimensionales se puede asegurar que si la función es invertible (logaritmo, función lineal, función cúbica…) el DP no se modifica, sin embargo si es no invertible (una función cuadrática) el DP puede quedar modificado. Es por ello que en (2) se puede expresar la toma de decisión bayesiana en términos del logaritmo de los elementos que la conforman. Una función monótonamente creciente asegura que dos scores de entrada con un determinado *ranking* (u orden) mantienen dicho *ranking* al aplicar la transformación. Por lo tanto si la transformación *s-LR* es de estas características se puede asegurar que el DP no se modifica.

Sin embargo, para transformaciones cuadráticas *s-LR* como la que se obtiene con un modelo gaussiano, si en la zona en la que se aplica la transformación presenta un crecimiento monótono, se puede asegurar que los LR de *scores* que caen en esta zona no presentan un cambio del DP. Por lo tanto interesará este tipo de condición para que en la etapa de generación de *s-LR* aseguremos que siempre que se mejora la calibración del sistema, no se estará empeorando el DP y por lo tanto siempre se mejorará el rendimiento global del sistema: recordemos que el rendimiento depende de ambos parámetros. Ello permitirá independizar la mejora de la calibración del rendimiento debido al DP.

3.5. Anchoring

El último concepto que se debe introducir es el *anchoring* o anclaje. Es un término relacionado con los criterios de generación de los *scores* de entrenamiento de los modelos *s-LR*. Esos *scores* se intentan generar para que las verosimilitudes de clases representen de la manera más adecuada el caso forense bajo estudio. El concepto de *anchoring* se entiende mejor con un ejemplo. Se tiene un sistema como el propuesto hasta ahora con dos clases Hp (*target*) y Hd (*non-target*). El problema es que para diferentes locutores las medidas de similitud obtenidas, *scores,* pueden variar, debido a la variabilidad intrínseca que presenta cada locutor en cuanto a sus condiciones fisiológicas (aunque los sistemas de reconocimiento de locutor tratan de reducir al mínimo cualquier tipo de influencia externa), (Doddington, 1998). Si el sistema compara locuciones e muchos locutores para generar *scores target,* cabe esperar que el rango de

dichas puntuaciones sea mayor que el que arrojaría si las locuciones pertenecen únicamente a un solo locutor, por ejemplo el sospechoso de el caso forense en cuestión. Esto mismo ocurre para puntuaciones *non-target*, aunque en mucha menor medida. Claramente, que los rangos de variación de *scores* sean mayores puede suponer una disminución del DP respecto a rangos menores (ver figura 3). Por lo tanto si se utiliza este modelo para calcular el LR en cuestión, se puede obtener peor rendimiento que si se utiliza un modelo adecuado al sospechoso, pues los datos presentarán menor variabilidad y por lo tanto los modelos pueden tener mejor DP, además de ajustarse más al sujeto en cuestión.

El *anchoring* por tanto no es más que una estrategia de generación de *scores* para entrenar los modelos de cálculo de LR más ajustada al caso forense en cuestión. Como, en general, el sospechoso es el mismo para cualquiera de las dos hipótesis en un caso forense, el anclaje al locutor (sospechoso) presente en el caso forense parece más adecuado.

Durante los años, (B. Hepler, P. Saunders, J. Davis, & Buscaglia, 2012), se han seguido numerosas estrategias de *anchoring*. Un ejemplo del mismo es el siguiente:

Supongamos que se tiene una locución obtenida de la escena del crimen, un sujeto sospechoso, y un conjunto de locuciones obtenidas a partir del sospechoso y por lo tanto de las que se conoce su identidad. Supongamos que en ese caso las clases (proposiciones) se definen de la siguiente manera:

- Hp (target) = La voz dubitada y la voz indubitada pertenecen a la misma persona y la voz indubitada fue generada por el sospechoso, y obtenida en unas condiciones acústicas determinadas.

- Hd (non-target) = La voz dubitada y la voz indubitada no pertenecen a la misma persona y la voz indubitada fue generada por el sospechoso, y obtenida en unas condiciones acústicas determinadas. Además la voz dubitada pertenece a un individuo de una población de autores potenciales, es decir, un individuo con características fisiológicas, dialécticas, socioelécticas…, parecidas a las del sospechoso.

Un esquema de *anchoring* sería fijar que las puntuaciones que modelan dicha clase deben contener una de las locuciones al menos perteneciente al sujeto sospechoso, tanto si Hp es cierta (*target*) como si lo es Hd (*non-target*). Esto es debido a que, en ambas proposiciones el sospechoso es el autor de la toma indubitada.

La definición de las hipótesis o proposiciones de ésta manera puede manifestarse en una falta de datos de entrenamiento debido a que el número de locuciones indubitadas tiende a ser escaso (imagine tener que tener al sospechoso durante horas grabando señal de voz). Por lo tanto, el número de puntuaciones *target* que se podrían generar para modelar la proposición Hp sería también escaso.

A lo largo de los años se han propuesto numerosas estrategias para aumentar el número de datos sobre todo de cara a entrenar la verosimilitud de la clase Hp, pues los modelos estadísticos clásicos de cálculo de LR (descritos más adelante) requieren de una gran cantidad de datos para su correcto funcionamiento. Una de las soluciones que se proponían era entrenar modelos con puntuaciones de *target* cualesquiera, estrategia que se ha llamado *no anclada* ya que no considera ningún sospechoso en particular. Sin

embargo, esta estrategia no se ajusta a la hipótesis definida por Hp tal y como se suele definir en casos reales, donde el sospechoso es conocido. Por ello, al no adaptar las verosimilitudes al locutor presente en el caso, cuya identidad es conocida, se perdía información. Otra de las formas propuestas para la generación de puntuaciones *target* era utilizar las locuciones dubitadas para la generación de puntuaciones. Esto no es correcto pues la identidad de la voz dubitada no se conoce. Por lo tanto las puntuaciones que se generen no se puede asegurar que sirvan para modelar ni la clase *target* ni la clase *non-target*, lo que por supuesto incumpliría el esquema de *anchoring* propuesto en este apartado.

4. Métodos de cálculo de *score-LR*

A continuación se van a presentar los métodos de cómputo de LR utilizados en el TFG. En general, salvo el modelo bayesiano, son métodos bien conocidos como Máxima Verosimilitud Gaussiana o Regresión Logística. El objetivo es describirlos cualitativamente, a excepción del enfoque bayesiano que propone usar este trabajo pues es necesario usar la matemática para poder entender la utilidad del mismo y la diferencia con respecto a los métodos clásicos. Además para la regresión logística y las *kernel density functions* se ha utilizado una librería ya existente para su implementación.

4.1. Regresión Logística

La regresión logística (en adelante RL o *Logistic-Regression*) es un modelo con un enfoque discriminativo pues tiene como objetivo obtener los parámetros de una transformación afín de un *score* a el log-ratio de las probabilidades a posteriori, presente en el exponente de (5), a través de la minimización de una función de coste. Como consecuencia, la regresión logística da lugar a una transformación *s-LR* en forma de recta entre el score de entrada y el log(LR) de salida.

La función de coste se obtiene a partir de una medida de error denominada Cllr (*likelihood ratio cost*). El Cllr se puede expresar de la siguiente manera, (D. van Leeuwen & Brümmer, 2007).

$$Cllr = \int_{-\infty}^{+\infty} Cdet(Pfr(\xi), Pfa(\xi), \xi) d\xi \quad (7)$$

Donde Cdet se expresa como, (D. van Leeuwen & Brümmer, 2007):

$$Cdet\big(Pfr(\xi), Pfa(\xi)\big) = Pfr(\xi) * \lambda fr * P(Hp) + Pfa(\xi) * \lambda fa * (1 - P(Hp)) \quad (8)$$

En esta expresión Pfr y Pfa hacen referencia a el porcentaje de decisiones erróneas de fa y fr sobre el total de decisiones tomadas. Según se mueve la frontera de clasificación (también llamado *frontera de decisión*), estos datos cambian y por lo tanto el error cometido, Cdet, dependerá de los parámetros del problema presentes en (8) . Por lo tanto el Cllr se puede interpretar como una medida promedio del error cometido para todas las posibles fronteras de decisión, dada la información independiente aplicación (LR); y los costes de decisión y probabilidades a priori, información dependiente de aplicación.

La solución a dicha integral es, (D. van Leeuwen & Brümmer, 2007):

$$Cllr(LR) = \frac{1}{2 * log2} * \left[\frac{1}{N_{Hp}} * \sum_{s=1}^{N_{Hp}} \log\big(1 + e^{-LLR(s)}\big) + \frac{1}{N_{Hd}} * \sum_{s=1}^{N_{Hd}} \log\big(1 + e^{LLR(s)}\big) \right] \quad (9)$$

Donde N_{Hp} y N_{Hd} representan el número de LR *target* y *non-target*, respectivamente y se utilizan para normalizar los resultados para poder ser comparados con conjuntos de LR de prueba con distinto número *target* y *non-target*. La regresión logística tiene como objetivo minimizar esta función de coste. Hay que recordar que en teoría de decisión bayesiana la combinación de costes de decisión e información a priori daba como resultado la frontera de decisión de Bayes, con la que se compara el LR para tomar decisiones. Por lo tanto tiene sentido que la solución a la integral para un conjunto de fronteras de decisión se exprese sólo en términos de LR, pues se promedian los posibles costes de decisión y probabilidades a priori, contenidos en la frontera de decisión (6). El Cllr por tanto es una medida de error que engloba el coste promedio para todos los posibles umbrales de decisión. A este coste le añade el coste debido a cómo el LR apoya las decisiones. Por lo tanto es una medida de error que incluye coste debido al DP y coste debida a la calibración.

4.2. Estimación puntual o de Máxima Verosimilitud

La estimación puntual o Máxima Verosimilitud (*Maximum Likelihood,* en adelante ML) es una manera diferente de enfocar el problema. En este caso el enfoque es generativo pues el objetivo es el de establecer distribuciones que modelen los datos, también conocidas como funciones de verosimilitud, $p(s|H,\boldsymbol{\theta})$. La diferencia básica con la regresión logística es que aquella genera directamente la transformación *s-LR* mientras que este tipo de enfoque genera distribuciones que mediante (3) se convierten en *s-LR*. Los apartados 4.3 y 4.4 son también enfoques generativos.

Antes de comenzar es necesario realizar la siguientes aclaraciones:

- Se asume que el conjunto de scores $\mathcal{S} = \{s_1, s_2, s_3, \dots, s_n\}$ para una de las clases es independiente del resto de las mismas por lo que el desarrollo se puede independizar de la clase en cuestión.

- Debido a la algorítmica típica de reconocimiento de locutores, se ha observado que el sistema de extracción de puntuaciones tiende a generarlas de manera gaussiana, sobre todo cuando las puntuaciones se generan para la hipótesis Hp. Esto es debido al uso generalizado de esquemas de normalización de scores (Navrátil & Ramaswamy). Por lo tanto, se ha escogido la curva gaussiana como modelo de los *scores*.

Dado que se puede independizar el cálculo de la verosimilitud de la clase se puede expresar dicha probabilidad mediante:

$$p\left(s\middle|\boldsymbol{\theta}, H = H_p\right) = p\left(s\middle|\boldsymbol{\theta}_p\right) (10)$$

El objetivo es asignar (10) a partir del conjunto de *scores* observados, es decir, calcular el valor del vector de parámetros que mejor representa la clase. Además se quitará el subíndice correspondiente a la clase del vector de parámetros, generalizando así el desarrollo para cualquiera de las mismas.

Así pues supongamos que tenemos un conjunto de datos $\mathcal{S} = \{s_1, s_2, s_3, \dots, s_n\}$ obtenidos a partir de una curva gaussiana parametrizada por $\boldsymbol{\theta}$ por lo que $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\mathcal{V}})$ donde $\boldsymbol{\mu}$

representa el vector de medias y $\boldsymbol{\nu}$ la matriz de covarianzas. Dado que los datos que se manejan son 1-dimensional $\boldsymbol{\theta}$ es un vector de parámetros restringido a una media y una varianza, $\boldsymbol{\theta} = (\mu, \nu)$.

Bajo esta suposición se puede expresar el conjunto de datos observados como:

$$p(\mathcal{S}|\boldsymbol{\theta}) = p(s_1, s_2, s_3, \dots, s_n|\boldsymbol{\theta}) = \prod_{k=1}^{n} p(s_k|\boldsymbol{\theta}) \quad (11)$$

Se ha aplicado independencia condicional debido a que en el momento en el que se conoce el vector que modela la distribución de los datos, dichos datos no aportan ninguna información adicional del modelo y por lo tanto son independientes entre ellos dado $\boldsymbol{\theta}$.

El objetivo de ML es el buscar los parámetros para maximizar (10), es decir, obtener el vector de parámetros $\boldsymbol{\theta_k}$ para el cual la función de verosimilitud toma su máximo valor para H_k.

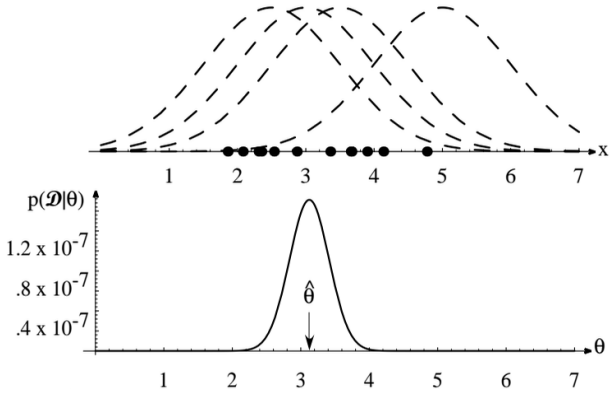


Figura 5: Representación de la función de verosimilitud. Nótese que en esta imagen D es $\mathcal{S}$ y x es s. (Duda, Hart, & Stork, Pattern Classification, 2000). En la figura superior se muestra un conjunto de datos observados junto a las posibles distribuciones que los modelan. El objetivo es encontrar cual de aquellas se asemeja más a los datos observados. En la figura inferior se observa la función de verosimilitud para los datos observados, función que da una idea de que valores de $\boldsymbol{\theta}$ representan mejor los datos observados.

La figura 5 ilustra el procedimiento de ML para una distribución gaussiana. Por lo tanto derivando la expresión de la función de verosimilitud y aplicando el logaritmo se pueden obtener los máximos mínimos y puntos de silla de la función. Mediante segundas derivadas se puede calcular cúal es el máximo y ese será el vector de parámetros escogidos, (Duda, E. Hart, & G. Stork, Pattern Classification, 2000).

Uno de los problemas de la estimación por ML es precisamente que se selecciona un único valor máximo como vector de parámetros del modelo. Notar que en el momento en el que se escoge $\boldsymbol{\theta} = \overline{\boldsymbol{\theta}}$ la suposición inicial de independencia condicional deja de ser cierta, ya que fijando un vector de parámetros se asume que se conoce un modelo que realmente no se conoce. Esto es debido a que, al no disponer de la población entera no

se puede asumir que los datos sean condicionalmente independientes del modelo. Otra forma de interpretar el problema es que se selecciona un vector de parámetros como si no presentara incertidumbre (cuando en realidad la presenta, ya que no se conoce), y a partir de ese momento se supone que el modelo viene definido por dicho vector. A este tipo de estimación se le conoce como estimación puntual, y por los problemas anteriores es incorrecta siempre, aunque cuando se dispone de muchos datos suele mejorar en robustez, pues la incertidumbre en el vector de parámetros se reduce.

Para el caso gaussiano la solución a dicha derivada para el caso particular de 1-d es:

$$\mu = \frac{1}{N} * \sum_{k=1}^{N} s_k \; ; \; \nu = \frac{1}{N} * \sum_{k=1}^{N} (s_k - \mu)^2 \quad (12)$$

Nótese que la estimación de la varianza ha sufrido un sesgo debido a que la media de la varianza no coincide exactamene con la varianza. Este sesgo desaparece según el número de datos tiende a infinito. Para más información consultar (Duda, Hart, & Stork, Pattern Classification, 2000) y (Peebles, 2006).

4.3. Modelado Bayesiano de Funciones Densidad de Probabilidad

El modelado bayesiano (en adelante MB) es una técnica generativa de asignación de verosimilitudes. En ciertas condiciones, y concretamente cuando se dispone de muchos datos (*scores*) para obtener las densidades, los resultados son iguales a los obtenidos usando ML. Sin embargo, la gran ventaja de este tipo de modelado es que permite mayor robustez que ML cuando hay escasez de datos. Esta técnica ha sido utilizada en reconocimiento de locutor recientemente, (Swart & Brümmer, 2014).

Ya se ha visto como la toma de decisiones bayesiana permite cometer el mínimo error en caso de conocer la función de verosimilitud de cada clase. También se ha visto que esta función salvo que se conozcan todos los datos de la población nunca va a ser conocida. La estimación bayesiana propone una estimación de la función de verosimilitud a partir de la denominada densidad de probabilidad predictiva, $p(s|\mathcal{S})$. En este caso se asume que nunca se va a conocer el vector de parámetros por lo que se promedian todas las posibles funciones de verosimilitud, es decir, todos los posibles valores de los parámetros (Figura 5). El objetivo es aproximar $p(s|\boldsymbol{\theta}, H)$ mediante la predictiva. Matemáticamente esto puede ser expresado de la siguiente manera:

$$p(s|\mathcal{S}) = \int_{\boldsymbol{\theta}} p(s, \boldsymbol{\theta}|\mathcal{S}) d\boldsymbol{\theta} = \int_{\boldsymbol{\theta}} p(s|\boldsymbol{\theta}, \mathcal{S}) * p(\boldsymbol{\theta}|\mathcal{S}) d\boldsymbol{\theta} = \int_{\boldsymbol{\theta}} p(s|\boldsymbol{\theta}) * p(\boldsymbol{\theta}|\mathcal{S}) d\boldsymbol{\theta} \quad (13)$$

Nótese que se ha aplicado la definición de marginalidad, la regla del producto y la independencia condicional para llegar a la expresión final. De nuevo se puede aplicar independencia condicional porque en el momento que se conoce $\boldsymbol{\theta}, \mathcal{S}$ no aporta ninguna información acerca de la función de verosimilitud que modela la clase ya que ella queda totalmente definida por el vector de parámetros. Además se puede observar la diferencia entre los modelados ML y MB. Mientras que ML trata de obtener el valor de $\boldsymbol{\theta}$ que maximiza la función de verosimilitud, lo que acaba derivando en una estimación puntual, MB supone que el modelo no se conoce y promedia todos los posibles valores

del vector de parámetros, de ahí que se pueda ver MB como una asignación promediada (de hecho como se verá más adelante efectivamente es un promedio).

A partir de (13) y dado que la función de verosimilitud con el modelo, $p(s|\boldsymbol{\theta})$, no es conocida se debe intentar modelar la probabilidad a posteriori del parámetro, $p(\boldsymbol{\theta}|\mathcal{S})$. Para ello entra en juego (11). Aplicando el modelado bayesiano se puede expresar la probabilidad a posteriori del parámetro como:

$$p(\boldsymbol{\theta}|\mathcal{S}) = \frac{p(\mathcal{S}|\boldsymbol{\theta}) * p(\boldsymbol{\theta})}{\int_{\theta} p(\mathcal{S}|\boldsymbol{\theta}) * p(\boldsymbol{\theta})\, d\boldsymbol{\theta}} \quad (14)$$

En este punto es interesante destacar que la suposición que permitía establecer (11) también puede ser expresado como:

$$p(s_1, s_2, s_3, \dots, s_n, \theta) = \prod_{k=1}^{n} p(s_k|\theta) * p(\theta) \quad (15),$$

aplicando la regla del producto. Por lo tanto (14) se puede interpretar a partir de (15) o de (11).

La interpretración de (14) es la siguiente. El denominador representa $p(\mathcal{S})$. En este caso se ha decidido expresarlo a través de la marginalización de la probabilidad conjunta expresada mediante (15), ya que pone de manifiesto que el denominador no es más que una constante de normalización, ya que supone dividir el numerador por su área. Por lo tanto no es más que un factor de escala que permite que la integral de la probabilidad a posteriori del parámetro, $p(\boldsymbol{\theta}|\mathcal{S})$, sea la unidad, tal y como establece el segundo axioma de la teoría de la probabilidad. La interpretación es exactamente igual que la que aplica en la toma de decisiones bayesiana: la diferencia radica en la información contenida en las variables aleatorias que toman parte. En este caso cualquier tipo de información a priori (es decir, antes de conocer los datos de entrenamiento) de la curva paramétrica está contenida en la probabilidad a priori del vector de parámetros, $p(\boldsymbol{\theta})$. Mediante la función de verosimilitud, $p(\mathcal{S}|\boldsymbol{\theta})$ se añade información de los datos generados a partir del modelo. Finalmente con la observación de los datos se obtiene la probabilidad a posteriori del parámetro, $p(\boldsymbol{\theta}|\mathcal{S})$.

Llegado a este punto conocemos la función de verosimilitud, $p(\mathcal{S}|\boldsymbol{\theta})$, su forma y aplicando (11) como calcularla. La forma es conocida porque es una decisión de diseño, una suposición del modelo. En nuestro caso, la función de verosimilitud se escoge como gaussiana, por las razones expuestas sobre los *scores* generados por el sistema biométrico (apartado 4.2) y por (Swart & Brümmer, 2014). Así pues el siguiente paso es cómo fijar la información a priori del parámetro. Llegados a este punto, a partir de aquí surgen dos vertientes: partir de una distribución no informativa, (Minka, 2001), o partir de una distribución denominada gaussian-gamma (en adelante GG), (Brümmer, 2011).

Una distribución no informativa se puede definir como una distribución uniforme en el intervalo $(-\infty, +\infty)$, es por lo tanto una distribución impropia, en el sentido de que su valor tiende a cero, pero su integral en todo el dominio es la unidad. Por temas de facilitar la comprensión se ha escogido la opción propuesta por Brümmer pero cabe destacar que ambas soluciones son válidas y a efectos prácticos equivalentes, tal y como expone Brümmer en (Brümmer, 2011).

Así pues la función gaussian-gamma se puede definir como, (Brümmer, 2011):

$$GG(\boldsymbol{\theta}|\boldsymbol{\Pi}) = GG(\mu, \nu^{-1}|\boldsymbol{\Pi}) = \mathcal{N}(\mu|\mu_0, (\beta * \nu^{-1})^{-1}) * Gam(\nu^{-1}|a, b) (16)$$

En esta expresión $\boldsymbol{\Pi} = (\mathrm{a}, \mathrm{b}, \mu_0, \beta)$ representa el vector de hiperparámetros y Gam es la distribución gamma con expresión dada por, (Brümmer, 2011):

$$Gam(\nu^{-1}|a, b) = \frac{b^a}{\Gamma(a)} * \nu^{-a+1} * \exp(-b * \nu^{-1}) \ (17)$$

$$\Gamma(a) = \int_0^\infty \nu^{-a+1} * \exp(-b * \nu^{-1}) \, d\nu^{-1} (18)$$

La Ecuación (18) es la expresión de la función gamma. La particularidad que tiene este tipo de función, (16), es que si escogemos la función densidad a priori del parámetro con esta forma y debido a que la GG es conjugada de la función gaussiana (escogemos la verosimilitud con esta forma, (Brümmer, 2011)), la probabilidad a posteriori del parámetro también es GG. Por lo tanto si hay solución analítica para (13), aunque se incorporen nuevos datos a la población, la probabilidad a posteriori del parámetro, (14), seguirá teniendo la misma forma y por lo tanto la solución para (13) seguirá teniendo la misma expresión analítica.

La forma que toma la probabilidad a priori del parámetro viene dada por el vector de hiperparámetros. Escogiendo dichos hiperparámetros adecuadamente se pueden obtener GG que tienden a ser no informativas. Por lo tanto (14) realmente se debe expresar como:

$$p(\boldsymbol{\theta}|\mathcal{S}, \boldsymbol{\Pi}) = \frac{p(\mathcal{S}|\boldsymbol{\theta}) * p(\boldsymbol{\theta}|\boldsymbol{\Pi})}{\int_{\boldsymbol{\theta}} p(\mathcal{S}|\boldsymbol{\theta}) * p(\boldsymbol{\theta}|\boldsymbol{\Pi}) d\boldsymbol{\theta}} \ (19)$$

En donde de nuevo la independencia condicional, entre el vector de parámetros y el vector de hiperparámetros, aplica a las funciones de verosimilitud, $p(\mathcal{S}|\boldsymbol{\theta}, \boldsymbol{\Pi}) = p(\mathcal{S}|\boldsymbol{\theta})$.

Finalmente, por suerte, hay solución analítica para (13). En dicha expresión e introduciendo en la integral el valor obtenido de (19) y conociendo la forma elegida de la función de verosimilitud (gaussiana), la integral del producto de una GG por una gaussiana tiene como resultado una densidad de probabilidad t-student con m grados de libertad según la siguiente expresión:

$$p(s|\mathcal{S}, \boldsymbol{\Pi}, H) = \frac{\Gamma(\frac{m+1}{2})}{\Gamma(\frac{m}{2})} * \frac{1}{\sqrt{\pi * \nu * (n+1)}} * \left(1 + \frac{(x-\mu)^2}{\nu * (n+1)}\right)^{-\frac{m+1}{2}} (20),$$

donde $n, m, \nu \ y \ \mu$ son parámetros obtenidos a partir de los hiperparámetros y los datos. Por tanto, son un conjunto de estadísticos suficientes con las siguientes expresiones, (Brümmer, 2011):

$$n = \beta + \sum_{k=1}^{N} s_k \, ; \qquad m = 2 * a + \sum_{k=1}^{N} s_k \ (21)$$

$$\mu = \frac{x1}{n}, x1 = \beta * \mu_0 + \sum_{k=1}^{N} s_k\,;\ \nu = \frac{x2}{n}, x2 = 2 * b + \beta * {\mu_0}^2 + \sum_{k=1}^{N} {s_k}^2 - n * \mu^2$$

Nótese que los estadísticos suficientes están formados a partir de los datos y los hiperparámetros. Por lo tanto, conociendo los datos $\mathcal{S}$ y fijando los hiperparámetros $\mathbf{\Pi} = (\mathrm{a}, \mathrm{b}, \mu_0, \beta)$, obtendremos la distribución predictiva de cada clase mediante (20) y (21).

Para finalizar este apartado se va a mostrar una gráfica con los resultados intermedios de todo el proceso, hasta llegar a la solución de la integral, viendo gráficamente como se refleja esta manera de inferir distribuciones en la transformación *score-LR* (Figuras 6 y 7).

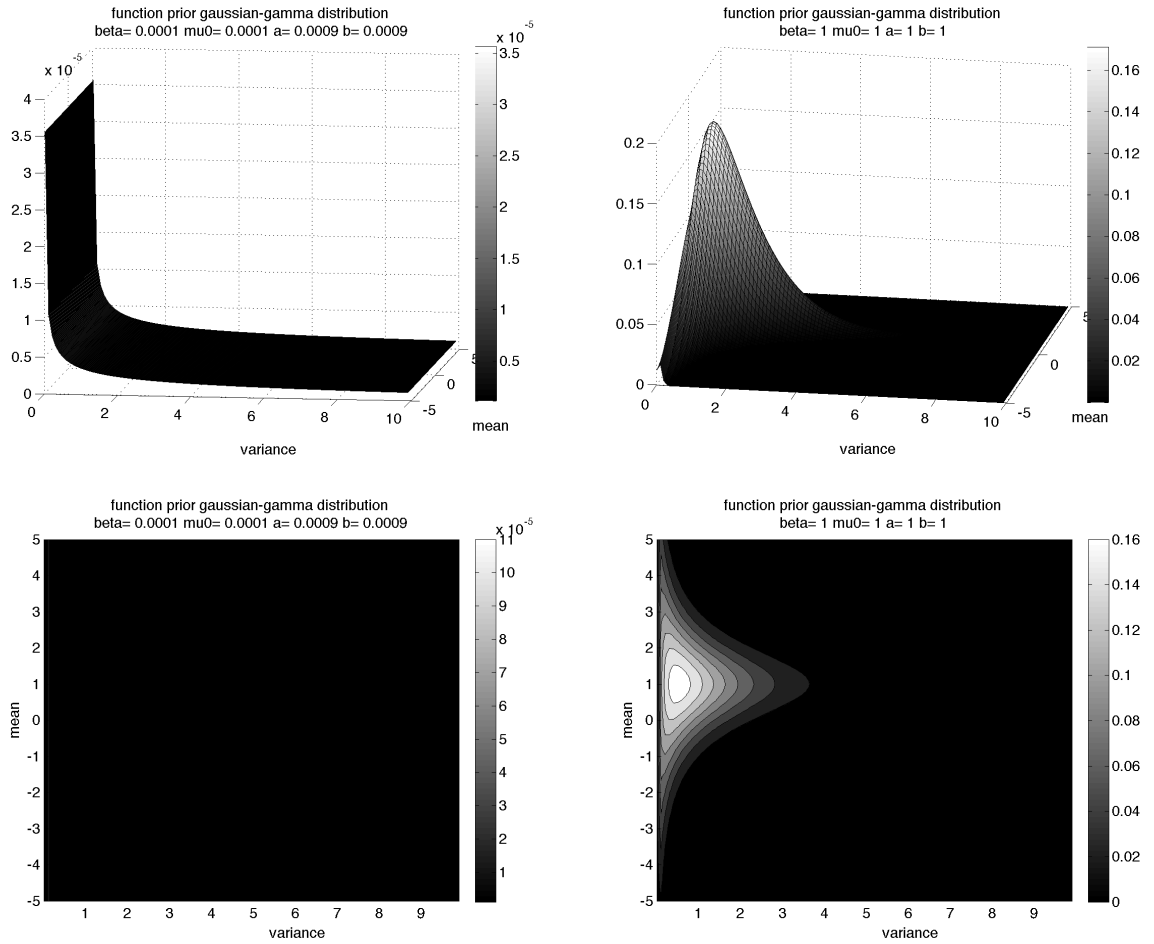


Figura 6: Gaussian-gamma prior distribution para dos combinaciones de hiperparámetros (indicados en el título de las figuras). A la izquierda se escogen hiperparámetros de manera que la GG tiende a ser no informativa. A la derecha hay una GG con unos hiperparámetros que incluyen información a priori del parámetro. Concretamente varianza en torno a 1 y media en torno a 0. Esta segunda función tendrá utilidad cuando se conozca información de la población a modelar antes de observar los datos.

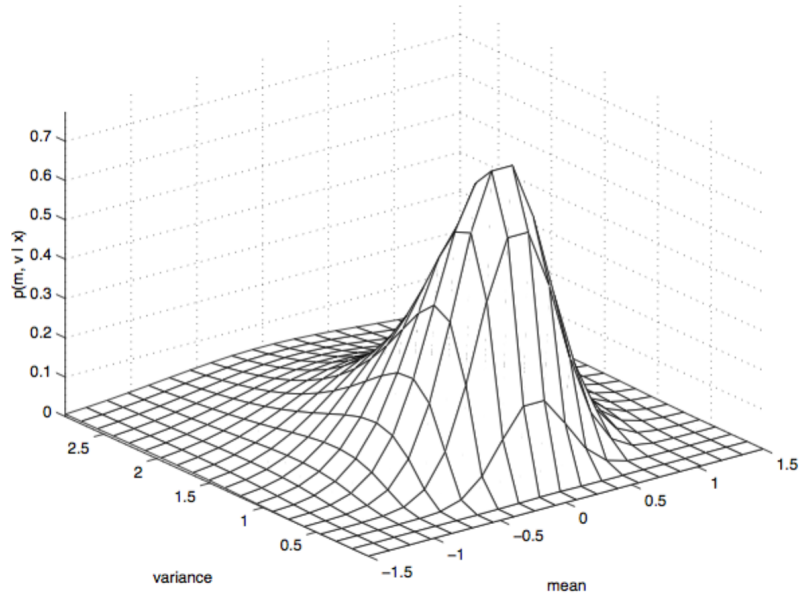


Figura 7: Esta figura representa la probabilidad a posteriori del parámetro. Imagen obtenida de (Minka, 2001).

Resulta interesante la gráfica que representa el producto de la GG por la función de verosimilitud normalizada por la integral (Figura 7). Se puede observar como tiene forma de GG, tal y como cabía esperar por la propiedad de conjugación. Además visto esto, si calculamos las marginales de la figura 7 se puede observar como la marginal de la media es una gaussiana y de la varianza es una gamma. Este es el método que propone Minka para la obtención de la predictiva, es decir, para calcular la solución de la integral, parte de las marginales de la probabilidad a posteriori obtenida mediante (19) y la función a priori del parámetro ya citada (no informativa). Es por ello que en (16) al formar la GG se condiciona la media de dicha distribución a ser descrita por una gaussiana y la varianza a una gamma (con lo que tiene sentido el enfoque que propone Minka en (Minka, 2001)). Sin embargo, cabe destacar que la solución para (13) es también una t-student con otros parámetros, ya que aquí no aplica la propiedad de conjugación. El desarrollo expuesto por (Minka, 2001) es válido para una selección de probabilidad a priori del parámetro de carácter no informativa.

De la figura 8 se puede sacar una conclusión interesante de todo el proceso de inferencia. La incertidumbre presente debido a la escasez de los datos se manifiesta en las colas de la t-student que al estar levantadas permiten incluir datos de un rango mayor que si se hubiese utilizado ML. Por otro lado cuanto mayor es la incertidumbre presente en los datos, más altas serán las colas que presenta esta t-student.

También, escoger una GG a priori no informativa aumenta la incertidumbre, ya que no se incorpora ninguna información a priori del parámetro, esto es, que media y varianza presentan los *scores* de entrenamiento lo que se manifiesta en que la t-student debe abarcar todavía mayor rango, para tener en cuenta que a priori no se sabe por dónde está la distribución.

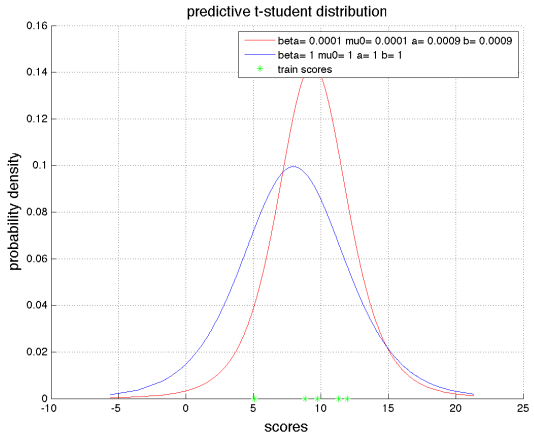


Figura 8: T-student para los dos tipos de hiperparámetros escogidos, para un conjunto de scores de entrenamiento mostrados en el eje x.

Dicho esto se puede observar en la figura 8 como no se cumple lo comentado en el anteriori párrafo, entonces: ¿por qué la t-student para la GG a priori (o *prior-GG*) menos informativa presenta mayor incertidumbre?. Esto es porque los datos a modelar no presentan media en torno a 0 y varianza en torno a 1 por lo tanto estamos incorporando información errónea del parámetro que los describe. Es por ello que para incluir esta falsa información la t-student debe tratar de abarcar el rango especificado por la *prior-GG*. Un ejemplo de esto sería el siguiente: imaginemos que tenemos un sistema que arroja datos de los que se sabe más o menos la media y la varianza que presentan, por lo que se fija una prior-GG que permita incorporar esta información, sin embargo los datos de entrenamiento de los que se disponen no son representativos de la distribución. Se puede observar como la información a priori permite modelar la información en torno a la media y varianza que más o menos se conoce y por lo tanto reducir el error ante nuevos datos de entrada que sí son representativos de la población. De alguna manera cuanto menos informativa es la información a priori, la t-student tiende a modelar la información que aportan los datos y sólo los datos.

En los experimentos que se realizarán más adelante supondremos que no se dispone de información a priori sobre los scores que calcula el sistema de reconocimiento de locutores. Por tanto, se utilizarán los hiperparámetros que hagan tender la *prior-GG* a una distribución no informativa, (Swart & Brümmer, 2014).

4.4. Modelo Bayesiano vs modelo de Máxima Verosimilitud

Para finalizar este apartado se va a mostrar la convergencia entre ML y el modelado Bayesiano terminando por concluir las ventajas que aporta el segundo.

Partiendo de (13), se puede interpretar aquella integral como un promedio estadístico con variable aleatoria dada por la función de verosimilitud y densidad de probabilidad dada por la probabilidad a posteriori del parámetro. En dicha densidad de probabilidad esta inmersa la información de los datos más la información a priori sobre la distribución de dichos datos. Lo que se realiza es un promedio de todas las posibles distribuciones que pueden modelar los datos, $p(s|\boldsymbol{\theta})$, y dicho promedio es la función predictiva que se utiliza para calcular los LR, manifestada en forma de t-student.

ML también puede ser explicado a través de (13). La diferencia es que ML escoge el parámetro que maximiza la función de verosimilitud. Esto puede ser equiparado a que la probabilidad a posteriori del parámetro se escoge mediante una delta de dirac bidimensional: $\delta(\boldsymbol{\theta} - \overline{\boldsymbol{\theta}})$. Matemáticamente esto puede ser expresado como:

$$\lim_{n\to\infty} \frac{1}{\sqrt{2 * pi * \nu}} * \exp\left(-0.5 * \sum_{k=1}^{n} (s_k - \mu)^2 * \nu^{-1}\right) = \delta(\boldsymbol{\theta} - \overline{\boldsymbol{\theta}}) (22)$$

A partir de (19) y aplicando la propiedad de selección de la delta a $p(\boldsymbol{\theta})$ se obtiene la probabilidad a posteriori del parámetro. Aplicando nuevamente la propiedad de selección de la delta en (13) la probabilidad predictiva es una gaussiana parametrizada por $\boldsymbol{\theta} = \overline{\boldsymbol{\theta}}$ **.**

Se puede ver como efectivamente cuando el número de datos tiende a infinito la inferencia mediante ML converge a la distribución real que modela los datos (Figura 9). El problema es que nunca se disponen de todos los datos y es ahí donde la inferencia ML comete el error comentado en la suposición de independencia.

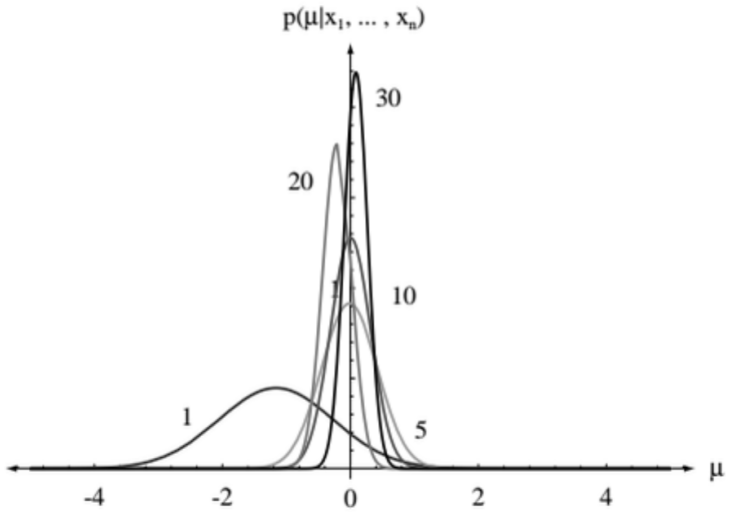


Figura 9: Probabilidad a posteriori del parámetro para el caso en el que se desconoce la media, (Duda, Hart, & Stork, Pattern Classification, 2000). Se puede observar la convergencia a una delta de dicha distribución según el número de datos aumenta a infinito. A partir de 50 datos de entrenamiento la t-student resultante converge a una gaussiana y la inferencia Bayesiana y ML generan la misma distribución predictiva a efectos numéricos.

Por lo tanto la estimación bayesiana será adecuada cuando el número de datos es pequeño ya que al promediar se reduce el "determinismo" que pueden inducir los datos de entrenamiento en la estimación puntual ML, y por tanto el error que esta última comete.

Finalmente se muestra una gráfica comparativa de ML vs Bayes y su transformación *s-LR* correspondi ente (Figura 10).

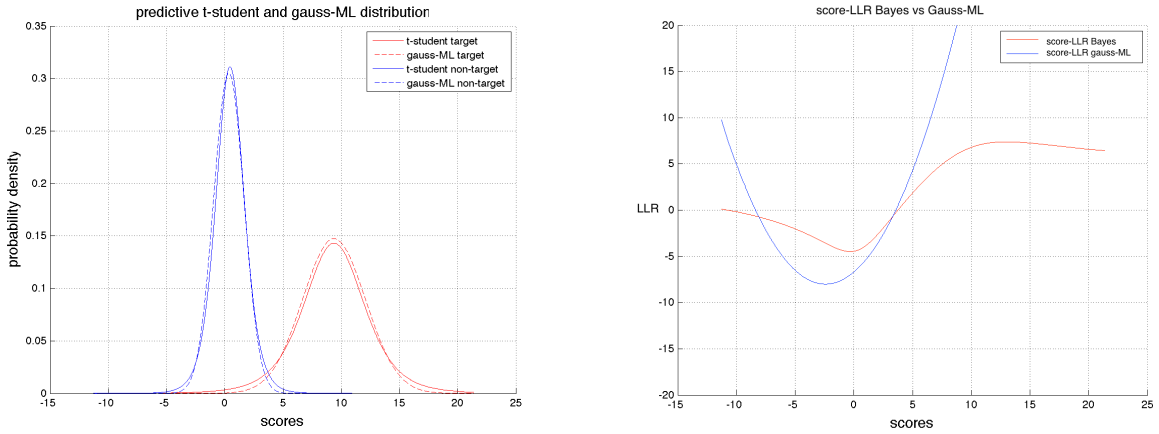


Figura 10: Modelado ML y Bayesiano para un conjunto de 5 datos de entrenamiento. Densidades de probabilidad (izquierda) y s-LR (derecha). Se puede observar como ante la escasez de datos Bayes permite asignar un modelo cuya transformación s-LR da lugar a valores de log(LR) mucho más moderados (eje y).

Efectivamente, en una transformación *s-LR* (3), el levantamiento de las colas se manifiesta en numeradores y denominadores que dan *s-LR* más moderados. Así pues la estimación bayesiana introduce información de incertidumbre de los LR mediante LR moderados en las zonas en las que apenas se disponen datos, o se ha establecido un parámetro a priori (como el ejemplo expuesto con anterioridad).

4.5. Kernel Density Functions

Este tipo de modelado, en adelante KDF, es un modelado no paramétrico, es decir, no puede representarse a partir de $\boldsymbol{\theta}$. A priori puede ser deseable cuando la distribución que presenta los datos no es ninguna distribución paramétrica.

De una manera muy simplista, este método se basa en el uso del denominado *kernel* que es una función con una forma parecida a alguna distribución paramétrica conocida, por ejemplo una normal. Lo que se hace es centrar este *kernel* en cada uno de los *scores* de entrenamiento ya que la finalidad de éste método es representar de la forma más parecida el conjunto de datos de entrenamiento.

Para una gaussiana el *kernel* queda definido de esta forma:

$$Gauss\_KDF\left(\frac{s-s_k}{h}\right) = \frac{1}{\sqrt[2]{2*pi}*h} * \exp\left(-0.5*\left(\frac{s-s_k}{h}\right)\right) (23$$

Donde h representa el ancho del *kernel* colocado en cada punto (nótese que la definición del *kernel* coincide con la definición de una gaussiana de media el *score* y desviación el ancho del *kernel*).

En la siguiente imagen queda de manifiesto el procedimiento de asignación de densidad.

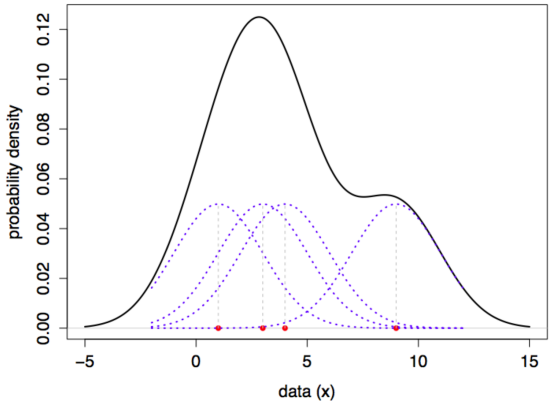


Figura 11: Asignación de densidad mediante gauss-KDF, (Zadora, Ramos, Martyna, & Aitken, 2014).

Matemáticamente el cálculo de la distribución puede definirse mediante:

$$p(s|H) = \frac{1}{N}\sum_{k=1}^{N} Gauss\_KDF\left(\frac{s - s_k}{h}\right) (24)$$

5. Bases de Datos y Medidas de Rendimiento

El siguiente capítulo tiene como objetivo describir los elementos utilizados para llevar a cabo el desarrollo del proyecto. Así pues se describirán las bases de datos utilizadas así como las medidas de rendimiento empleadas.

5.1. Bases de Datos

Para el desarrollo de este proyecto se han utilizado dos bases de datos. Ambas pertenecen a campañas de evaluación de tecnologías de reconocimiento de locutores organizadas por el National Institute of Standards and Technology americano. Dichas evaluaciones son abiertas a cualquier participante, ciegas (en el sentido de que nadie conoce los verdaderos resultados de la evaluación hasta después de la misma) y utilizan bases de datos y protocolos de evaluación comunes. Por tanto, se han convertido en *benchmarks* estándar de facto en las tecnologías de reconocimiento de locutores.

Para la primera parte se ha utilizado un conjunto de *scores* obtenidos a partir de la evaluación NIST Speaker Recognition Evaluation (SRE) 2012. Dicha base de datos está formada por un conjunto variado de locuciones adquiridas en distintas condiciones de grabación. En todas las locuciones, el habla es conversacional telefónica, salvo en el caso de locuciones de entrevistas, en las que el locutor es entrevistado de forma presencial. Factores como el idioma, acento, entorno acústico, canal de transmisión, estado anímico y esfuerzo vocal varía enormemente de una locución a otra, lo que da lugar a una tarea de reconocimiento muy desafiante.

Se enfrenta siempre una locución de 2 minutos de media (llamada *de test*) con un conjunto de locuciones (llamadas *de modelo*) cuya cantidad depende para cada locutor. Ello da lugar a un total de 63.666 comparaciones *target* y 13.877.747 comparaciones *non-target*, que forman una base de datos de scores llamada *total*. Existen locuciones de hombres (*male*) y de mujeres (*female*), pero se comparan siempre locuciones de locutores del mismo género. Además, dependiendo del tipo de habla que se compara, se da lugar a diferentes *condiciones de evaluación*, que en este trabajo se referirán como sub-bases de datos de scores de la base de datos de scores total. Estas condiciones suceden tanto para hombres como para mujeres, y son las siguientes:

- **Phone Tel Phone Tel:** Esta primera condición todas las locuciones están obtenidas a partir de dos locutores hablando a través de un teléfono móvil. En la memoria se resumiran como PTPT. En total se generan 3.533 LR *target* y 191.220 LR *non-target*.

- **Phone Tel Phone Mic:** Esta segunda condición las locuciones de modelo están obtenidas una a partir de un telefono móvil y la de test en una conversación telefónica grabada a través de un micrófono. Existen varios tipos de micrófono que varían de locución a locución. En la memoria se resumirán como PTPM. En total se generan 430 LR *target* y 24.207 LR *non-target*.

- **Phone Tel Int Mic:** Las locuciones de modelo están obtenidas a partir de un sujeto hablando a través de un teléfono móvil y en la de test el locutor está

hablando a un micrófono en una entrevista. En la memoria se resumirán como PTIM. En total se generan 1801 LR *target* y 59.583 LR *non-target*.

- **Int Mic Int Mic:** Todas las locuciones pertenecen a usuarios hablando a través de dos micrófonos en una entrevista. En la memoria se resumirán como IMIM. En total se generan 970 LR *target* y 27.283 LR *non-target*.

La base de datos NIST SRE 2012 se utilizará como base de datos de validación, o *test*, para probar diferentes esquemas de cálculo de LR. Para el entrenamiento de los modelos de LR, se utilizará la base de datos NIST SRE 2010, tal y como el grupo ATVS hizo cuando se presentó en la evaluación.

El sistema utilizado en ambas evaluaciones en un sistema de cálculo de puntuaciones que no es objetivo de este TFG, y que utiliza una tecnología i-Vector PLDA. Los detalles se pueden consultar en (Stafylakis, Kenny, Ouellet, Perez, Kockmann, & Dumouchel, 2013)

Para la simulación de casos reales se utiliza la base de datos y protocolo de la evaluación NIST *I-vectors Challenge*. En esta evaluación, NIST provee a los participantes de un i-Vector por locución, que es una representación vectorial de las características más relevantes de la locución en cuestión. Con estos i-Vectors, se utilizarán los mismos para calcular los *scores* de comparaciones de i-Vectors. Todos los i-Vectors son de locutores masculinos y habla telefónica. De nuevo, el sistema utilizado no es objetivo de este TFG, y está descrito en (Lozano-Diez, Gomez-Piris, Franco-Pedroso, Gonzalez-Dominguez, & Gonzalez-Rodriguez, 2014), y cuenta con etiquetas de locutor para los i-Vectors de desarrollo utilizados para entrenar el sistema de cálculo de puntuaciones.

Los *scores* generados por el sistema del i-Vector Challenge se utilizarán para simular casos reales y para probar el modelo bayesiano propuesto en diferentes situaciones de anchoring. El protocolo utilizado se describirá en la parte experimental correspondiente a dicha simulación.

5.2. Curva Det (Detection Error Tradeoff)

La curva Det es una medida del DP presente en dos poblaciones. No es una medida de rendimiento del clasificador, pues mide el DP, y en el caso de cálculo de LR es necesario medir también la calibración. Sin embargo pese a que este trabajo está centrado en el nivel de presentación, es necesario conocer el DP de la población con la que se va a trabajar, pues recordemos que influye notablemente en el rendimiento.

Esta curva mide Falsos Negativos (*False rejection*), es decir aquellos *scores* clasificados como *non-target* cuando son *target*; y Falsos Positivos (*False acceptance*): aquellos clasificados como *target,* cuando son *non-target*. Dado que estos valores dependen del umbral de decicsión, la curva DET representa ambos valores para todos esos umbrales.

Para ello hace uso de (8) en donde se va variando el umbral, dado por (6), desde menos infinito a más infinito. Se evalua el coste mediante Pfa y Pfr, que miden el número de datos mal clasificados cuando la frontera de decisión está situada en un lugar determinado (ese lugar simulará costes asociados a los fallos y probabilidades a priori ya que recordar que (6) depende de ellos).

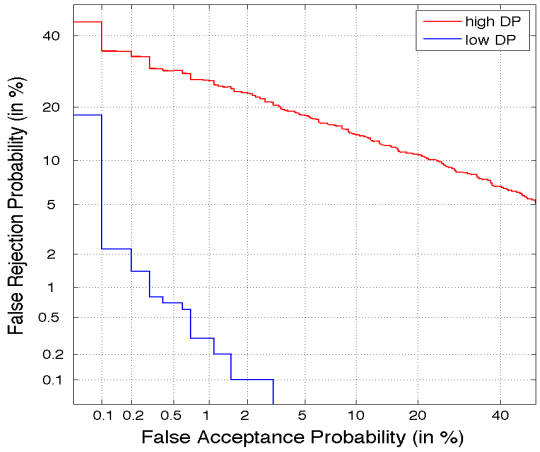


Figura 12: Ejemplo de curva Det. Cuanto más cerca está la curva al origen de coordenadas más DP presenta la población. Este es el DP correspondiente a las distribuciones de la figura 3, si bien la de la izquierda tenía DP máximo, se ha modificado ligeramente la media de las distribuciones para que haya algo de DP.

Por lo tanto debido a (8) tiene sentido que cuanto mayor solape tienen las distribuciones más Pfa y Pfr habrá para diferentes umbrales y por lo tanto más coste, reflejado en un peor DP. Por lo tanto dos distribuciones presentaran mismo DP cuando presenten misma curva DET. Otra forma de verlo es: dos distribuciones presentan el mismo DP cuando se puede encontrar una frontera en cada una de las distribuciones, que no tiene por qué ser la misma, para la que se tenga el mismo valor de Pfa y Pfr, pues este depende del grado de solape que es lo que caracteriza al DP.

Más información acerca de esta medida puede ser encontrada en (D. van Leeuwen & Brümmer, 2007), (Martyna, Zadora, Ramos, & Aitken, 2014), (Ramos , Forensic Evaluation of the Evidence Using Automatic Speaker Recognition Systems, 2007).

5.3. Curva ECE (*Empirical Cross Entropy*)

La curva ECE es una medida a partir de la cual se puede obtener tanto el coste asociado al DP como el coste asociado a la calibración.

El fundamento básico de la curva ECE es calcular la pérdida de información de una variable debido a incertidumbre. En el contexto forense, ECE es la información que se pierde sobre las proposiciones de un caso forense debido a la incertidumbre acerca de las proposiciones. Como esa información perdida depende de las probabilidades a priori, ECE se suele representar como curva ECE, en función del logaritmo del ratio de probabilidades a priori $\log\left(O(Hp)\right) = \log\left(\frac{P(Hp)}{P(Hd)}\right)$ (Figura 13).

Se puede demostrar que ECE es igual a Cllr, definido en (9), cuando la probabilidad a priori es 0.5. Es decir, en una curva ECE se visualiza Cllr en el punto de cruce de la curva con el valor 0 del eje de las x.

Cabe recordar que Cllr incluye el rendimiento debido tanto al DP como la calibración, y de la misma manera ocurre con ECE.

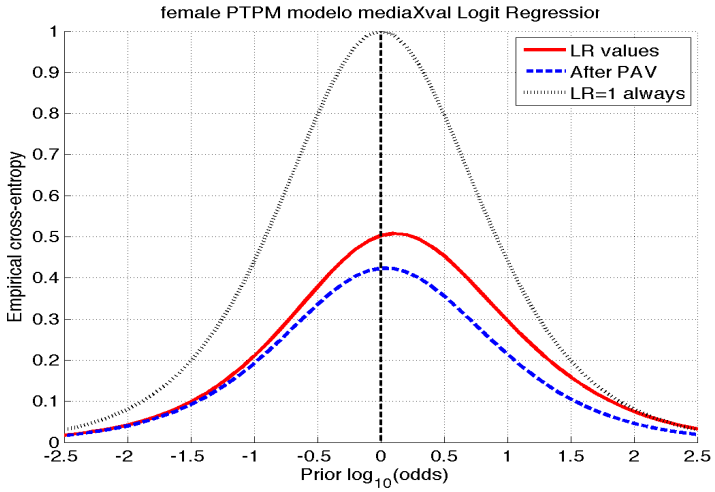


Figura 13:Ejemplo de curva ECE. La línea roja mide el rendimiento total (cuando menor, mejor) y la azul el rendimiento debido al DP (cuanto menor, mejor DP). Por lo tanto la diferencia entre ambas es la pérdida de rendimiento debido a la falta de calibración.

El cálculo de ECE en cada punto queda descrito mediante la siguiente expresión, (Gonzalez-Rodríguez, Ramos, Zadora, & Aitken, 2012):

$$ECE = \frac{P(H_p)}{N_p} * \sum_{s=1}^{N_{Hp}} \log_2\left(1 + \frac{1}{LR(s) * O(Hp)}\right) + \frac{P(H_d)}{N_{Hd}} * \sum_{s=1}^{N_{Hd}} \log_2\left(1 + LR(s) * O(Hp)\right) \quad (25)$$

Nótese que en caso de que las probabilidades a priori sean 0.5 la expresión dada por (25) y por (9) son iguales.

Por otro lado, se puede demostrar, que el Cllr se puede dividir en el Cllr asociado al DP llamado minCllr, y el Cllr asociado a la calibración, llamado calCllr, (D. van Leeuwen & Brümmer, 2007). Mediante el algoritmo de PAV, más detalles en (D. van Leeuwen & Brümmer, 2007) se puede extrar el minCllr asociado al DP. Por lo tanto el calCllr debido a la calibración es la diferencia entre el Cllr (9) y el resultado que arroja el algoritmo de PAV. De la misma forma, mediante el algoritmo PAV se puede dividir la ECE total en la minECE debida a la pérdida de información por un DP imperfecto, y la calECE, debida a la pérdida de información por una calibración imperfecta.

Si una transformación *s-LR* es invertible (por lo tanto todas las funciones monótonamente crecientes están incluidas), sabemos que tanto minECE (la curva azul en la curva ECE) como minCllr no cambiarán, y se podrá independizar el estudio del rendimiento del coste asociado al DP o lo que es lo mismo, dado un DP podremos entrenar modelos con el fin de mejorar la calibración independientemente del DP.

La curva ECE es, como se deduce, una generalización del Cllr, que permite un análisis más detallado del comportamiento del método de LR en términos de la teoría de la información. Por otra parte, Cllr presenta la ventaja de ser un valor escalar, y por lo tanto ser útil como medida única resumida de rendimiento, y como resumen de una curva ECE. Ambas medidas de rendimiento se utilizarán en este TFG.

Más información acerca de la curva ECE en (D. van Leeuwen & Brümmer, 2007), (Ramos , Forensic Evaluation of the Evidence Using Automatic Speaker Recognition Systems, 2007) (Ramos & González-Rodriguez, Reliable support: Measuring calibration of likelihood ratios, 2013) (Gonzalez-Rodríguez, Ramos, Zadora, & Aitken, 2012) (Martyna, Zadora, Ramos, & Aitken, 2014) (Ramos, Fiérrez, Krish, & Meuwly , 2015).

6. Experimentos y Resultados

Una vez expuesto el procedimiento de toma de decisiones en un caso forense, las bases de datos y las medidas de rendimiento el siguiente paso es el de hacer uso de las técnicas expuestas con anterioridad para poder analizar diferentes problemas.

Este apartado se va a dividir en una parte dedicada a observar efectos en la parte de entrenamiento y otra para la parte de validación. Finalmente se mostrará un escenario de *anchoring* para realizar la comparación entre las técnicas de gauss-ML y gauss-MB permitiendo verificar la ventaja expuesta en el apartado de medidas de cálculo de LR.

Además se expondrá la técnica de validación cruzada utilizada en la parte de entrenamiento, y se realizará un análisis de diferentes problemas y diferentes hipótesis para poder solucionarlos.

La manera en la que se va a proceder es la siguiente: primero se presentará un método para ver el desajuste que presentan los datos de entrenamiento. Posteriormente se hace uso de las curvas ECE para medir el DP y la calibración de los métodos de transformación *score-LR*: Regresión Logística, gauss-ML y gauss-KDF. Además se medirá el grado de sobreentrenamiento que presentan los modelos analizando y descartando o validando su uso. Finalmente se analizará la bondad del cálculo de LR con un conjunto de datos de *test*.

Los dos primeros apartados van orientados a definir la técnica expuesta para analizar el desajuste y la variabilidad de los datos fruto de este desajuste.

6.1. Variabilidad de un conjunto de datos

En un problema de cálculo de LR las técnicas clásicas requerían de muchos datos de entrenamiento para funcionar correctamente. Aun así, distintos conjuntos de datos, incluso perteneciendo a la misma base de datos, pueden dar lugar a modelos de descripción de los mismos diferentes.

Por lo tanto la variabilidad que puede presentar una base de datos con la que se entrena un modelo puede resultar un problema si no se analiza adecuadamente. Además otro problema que se puede encontrar es el sobreentrenamiento, causa debida al modelo, pero que a la hora de medir rendimiento se puede manifestar como una variabilidad de dicho rendimiento dentro del conjunto de datos, como se mostrará al mostrar el conjunto de pruebas realizadas. Así se puede definir:

- **Sobreentrenamiento:** Es el fenómeno por el que un modelo se ajusta tanto a los datos de entrenamiento que acaba por no representar bien el conjunto entero de la población, ya que los datos de entrenamiento son siempre incompletos, y además pueden presentar los denominados *outliers* que son datos extraños a la población y por lo tanto no representativos de ella. Este tipo de datos extraños pueden surgir por múltiples motivos y no es objetivo de este trabajo realizar dicho análisis.

- **Desajuste de los datos:** Dentro del desajuste de datos podemos encontrar desajuste entre los datos de *train* y los de *test*. Este desajuste ocurre cuando los datos de entrenamiento representan bien la población a la que pertenecen (posteriormente se describirá en más detalle qué quiere decir esto), sin que se haya utilizado un modelo sobreentrenado, pero el rendimiento obtenido con los datos de *test* no es el adecuado. Además se puede encontrar desajuste dentro de los propios datos de *train*, presente cuando los propios datos de entrenamiento no son adecuados para generar el modelo. Este problema se está investigando en la actualidad y no es objetivo de este trabajo. Este desajuste de base de datos causa una disparidad entre modelos de entrenamiento y de test. Algunas de las causas de estos efectos se describen a continuación:

    - **Complejidad del Modelo:** El numero de datos no es suficientemente para representar el modelo que se quiere entrenar. Esto ocurre para modelos complejos. En general con un modelo gaussiano no suele haber problema con el conjunto de datos manejados. En definitiva, hay modelos más robustos que otros ante la falta de datos.

    - **Presencia de subgrupos:** El conjunto de datos presenta subgrupos debido a la presencia de variabilidad en la toma de las locuciones, por ejemplo distintos tipos de micrófonos.

    - ***Outliers*:** Presencia de datos no representativos de la población. Su naturaleza puede ser por ejemplo particularidades de las locuciones de las que se saca la puntuación, locuciones excesivamente cortas, etc.

    - **Desajuste de datos de entrenamiento:** Para entender este último concepto se va a dedicar el siguiente apartado.

6.1.1. Variabilidad de los datos.

En este subapartado se va a describir el problema de variabilidad presentado en este trabajo. Cabe destacar que la variabilidad de los datos es un concepto mucho más amplio dentro de la clasificación de patrones ya que puede presentarse debido a un mal diseño del proceso de extracción de características o de cálculo de *score* en el nivel de discriminación, o por problemas intrínsecos a los datos que se manejan (en voz, por ejemplo, la variabilidad está siempre presente de forma muy significativa).

Para ello introducimos el siguiente ejemplo. Supongamos que conocemos el rango en el que queda definida la fdp de una variable aleatoria continua que describe los datos que queremos modelar. Lo ideal sería tener muchos *scores* observados dentro del dominio de la fdp para modelar de manera adecuada la forma de dicha distribución. Por lo tanto se va a asumir que se tiene un número suficiente de *scores* para modelar la forma de la fdp, pero ello no implica que esa fdp obtenida sea la adecuada.

El primer problema que puede presentarse ya se ha comentado y es el debido a los *outliers*. Por otro lado dentro del conjunto de *scores* puede ocurrir que los datos de los que se disponen no son representativos de la variable aleatoria

citada. Este efecto (llamado en estadística *sampling effect*) está ilustrado en la Figura 14:

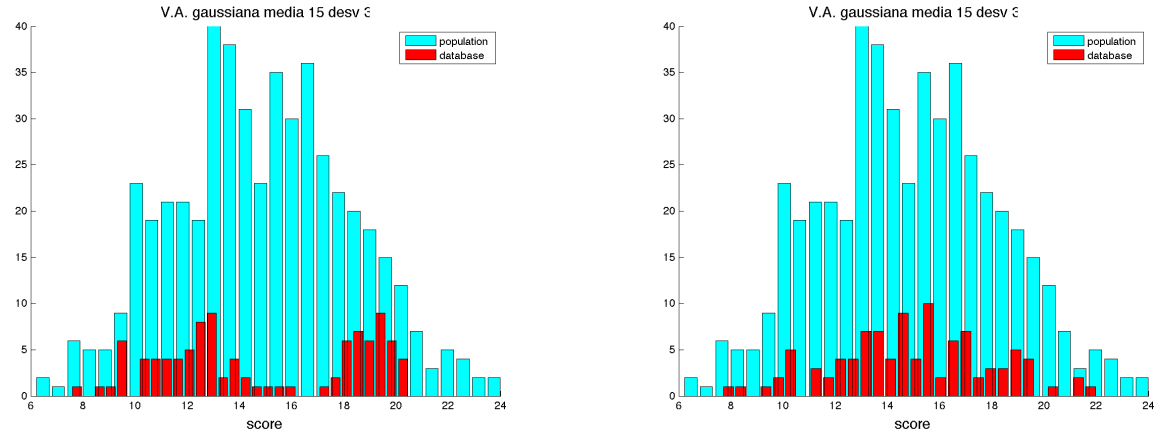


Figura 14: Histogramas presentando el problema de variabilidad analizado. En azul se representa la población, generada con un modelo gaussiano, e igual en ambas gráficas. En rojo la base de datos de la que se dispone para el entrenamiento del modelo, diferente cada vez que elegimos un conjunto de datos de entrenamiento.

A la vista de la Figura 14 se puede intuir como efectivamente pese a que el número de datos de la base de datos es suficiente para modelar una gaussiana la base de datos de la izquierda es menos representativa de la población general, dado que los datos seleccionados se ajustan mucho menos a la gaussiana, con la que se generaron los datos de la población. La variabilidad es una propiedad de la variable que genera los datos por lo que la variabilidad de ambas figuras es la misma. Sin embargo debido a este efecto de variabilidad la población representada por la base de datos no es siempre la adecuada. Por lo tanto para la base de datos de la derecha, las diferentes muestras (bases datos) de la población total (debido a que el conjunto de *scores* está suficientemente aleatorizado entre cada división) obtendrán modelos parecidos y representativos de la población por lo que se obtendrán en general rendimientos adecuados y similares, salvo pequeños problemas con *outliers*.

Por otro lado en la figura de la izquierda se puede observar como la base de datos no es representativa de la población en general, esto es debido a que los datos presentan una variabilidad que da lugar a una base de datos poco representativa (no son *outliers* ya que no son datos que se alejen de zonas de la distribución, son datos pertenecientes a zonas en las que la distribución está definida). Este efecto se manifiesta como un grado de desajuste *train-test* (que será analizado y visualizado más adelante) como para el propio desajuste de los datos de *train*. En los datos de *train* se podrá apreciar porque se entrenaran modelos con los que al medir rendimiento se verán afectados por datos pertenecientes a una zona diferente del rango. Cabe destacar que con dichos datos puede seguirse modelando una gaussiana (aunque la selección realizada ha sido con el fin de mostrar el problema por lo que se pueden apreciar dos pequeñas gaussianas,

pese a que los datos son una selección de los pintados en azul) por lo que el problema se debe a que los propios datos no representan el conjunto al que pertenecen. Un detalle importante: la población representada no es la población general, a la que pertenecerían los datos de *test* también sino la población que los datos de *train* pretende representar. Puede producirse desajuste *train-test* y no apreciarse en la parte de entrenamiento, algo que se observará cuando se analice el rendimiento de las técnicas propuestas con un conjunto de *scores* de *test*.

6.2. Validación cruzada

La validación cruzada es una técnica utilizada para un uso óptimo de unos datos para reflejar rendimientos fiables. En este TFG además la utilizaremos para analizar el desajuste presente en los datos de *train*. Dicho desajuste puede verse manifestado en las diferentes causas expuestas en el apartado anterior por lo tanto la correcta interpretación de los resultados resulta de máxima relevancia para analizar la naturaleza del problema. Por ejemplo podríamos pensar que la base de datos está desajustada cuando es el modelo que genera sobreentrenamiento el causante del mal rendimiento.

La validación cruzada consiste en dada una base de datos que representa un conjunto de *scores* de la clase Hp y un conjunto de *scores* de la clase Hd, subdividir la base de datos en K subbases, entrenando el modelo con K-1 bases de datos y realizando una prueba de test con la subdivisión restante.

Para este trabajo se ha utilizado un valor de K=4. Además cuando a lo largo del texto o en una figura se vea Database123, ello implicará que se han utilizado las subbases 1,2,3 para entrenar el modelo y la subbase 4 para hacer el test. La subdivisión de las bases de datos se realiza siempre de la misma manera, se cogen los datos aleatorizados y se subdividen en 4 conjuntos de *scores* seguidos.

Dicho esto, la validación cruzada tiene dos grandes objetivos:

- **Detectar sobreentrenamiento:** Si se dispone de un modelo y se entrena con una subdivisión de la base de datos y al realizar la prueba de *test* el rendimiento obtenido no es adecuado y esto ocurre realizando todas las combinaciones posibles (para k=4 serían 4: Database123, Database124, Database134 y Database234) es posible que el modelo tienda a sobreentrenar.

- **Detectar desajuste dentro de los datos de train:** Analizar la variabilidad presente en los datos de train. Esto se observará cuando dado un modelo del que sabemos que no tiende a quedar sobreentrenado, si se produce un mal rendimiento ello puede deberse o bien a que la subbase empleada en *test* presenta toda ella un conjunto de *outliers* (en cuyo caso obtendremos mal rendimiento para una de las subdivisiones y además es algo poco probable) o bien que los datos no son representativos de la población y el modelo no es capaz de corregir este desajuste (obtendremos rendimientos generalmente poco adecuados para las k subdivisiones en comparación a otros rendimientos obtenidos con el mismo algoritmo y otras bases de datos). Ello es debido a que (ver figura 14 izquierda) las diferentes subdivisiones contendrán datos de diferentes zonas lo que hará que haya desajuste entre las bases de datos de entrenamiento y la utilizada en validación. Se puede ver

como que las diferentes subdivisiones contendrán datos tanto de la primera gaussiana como de la segunda.

Además, estos efectos pueden ser observados en la represntación de las fdp de los modelos y las *s-LR*. EL uso de curvas ECE permitirá comprobar lo que a priori se puede intuir de los modelos y las *s-LR*.

6.3. Análisis de los modelos en los datos de entrenamiento

El primer paso es mostrar las distribuciones entrenadas. Debido a que MB y ML en condiciones de muchos datos arrojan los mismos resultados (ver apartado correspondiente) no se va a utilizar el esquema bayesiano.

Además se mostrarán las transformaciones *score-LR* a las que dan lugar los diferentes modelos. La Regresión Logística por ser un enfoque discriminativo no da lugar a fdp de los datos.

En las figuras expuestas en el anexo A se muestran además el resultado enmarcado en el procedimiento de validación cruzada para los 3 algoritmos. Hay varios aspectos interesantes que resaltan a la vista observando las transformaciones *s-LR* y los modelos.

A partir de ahora el objetivo es ir relacionando la problemática introducida en los dos apartados anteriores con lo observado. Posteriormente se presentarán las curvas ECE para terminar de comprobar los resultados.

En la transformación de score a log(LR) (LLR), el modelado gaussiano da lugar a curvas *s-LLR* polinómicas de orden 2, mientras que la Regresión Logística da *s-LLR* lineales y KDF, por ser un modelo no paramétrico, dependerá de los datos la forma de la transformación.

Por otro lado, todos los modelados presentan cierta variabilidad en cuanto a los modelos generados, es decir, para cada combinación se obtienen curvas que van variando, tal y como se puede observar en las figuras del Anexo A (en adelante se usará A.1 para referirse al apartado 1 de dicho anexo). Cabe destacar que donde más se aprecia esta variabilidad es en KDF (ver A.3) lo que lleva a la conclusión de que es un modelo que parece que se sobreajusta demasiado ya que si la variabilidad se debiese a los datos, todos los modelos presentarían algo más de diferencia entre los modelos entrenados para las diferentes subdivisiones. De hecho, se pueden observar picos en las transformaciones *s-LR* fruto del ajuste al conjunto de datos. Los cambios de curvatura en el modelo se manifiesta de esta manera por lo tanto no interesan *s-LR* con cambios tan bruscos en el apoyo de la decisión para *scores* tan cercanos, no tiene ningún sentido. Todo hace indicar que KDF es un modelo que presenta el problema de **sobreentrenamiento.**

Por otro lado la Regresión y ML presentan mucha menos variabilidad. Sin embargo, la subdivisión correspondiente a la línea punteada roja en los modelos y la *s-LR* verde siempre es diferente al resto lo que indica que esa subdivisión de la base de datos presenta ligera variación, lo que puede indicar o bien que hay una subdivisión no representativa de la población (figura 14, izquierda) o bien que hay un conjunto de *outliers*. De las dos opciones parece más probable la segunda debido a que una población no representativa presentaría algo más de variación entre los modelos (se

observa como las otras tres divisiones presentan modelos muy parecidos). La base de datos PTPT podría presentar una población no representativa. Como se ve, se obtienen cuatro curvas *s-LR* diferentes lo que indica que los datos presentes en la base de datos representan curvas distintas. A priori, dado que el modelo se entrena con todos los datos (no se escoge una subdivisión para entrenar), esta variabilidad puede compensarse al juntar todos los datos dando lugar a un modelado correcto, es decir, observando A.1 PTPT se podría ver como que la gaussiana entrenada con todos los datos engloba las diferentes gaussianas generadas en el proceso de validación cruzada, una especie de media entre las 4 gaussianas. Sin embargo esto no es más que una suposición que habrá que comprobar. Lo que se ha observado es un indicativo de que esa base de datos presenta problemas.

Además, dado que los *scores* de entrenamiento *target* son más escasos que los de *non-target*, las curvas *target* (rojas) presentan mucha mayor variabilidad (ver A.1) que las azules. Ello también es consecuencia de lo comentado en el apartado 6.1.1. y se debe al **desajuste de los datos de train**, no tanto a la presencia de *outliers*. Al fin y al cabo si el sistema biométrico es suficientemente robusto, los *outliers* presentes no serán tantos como para manifestarse en un rendimiento negativo. En la validación cruzada se pueden observar si tenemos la mala suerte de seleccionar una subbase de datos que contenga solo *outliers*.

Por lo tanto la primera tarea es escoger qué modelado se va a utilizar para la parte de *test*. Ha quedado de manifiesto que interesan curvas lo más estables posibles para que a la hora de generar el modelo garanticemos que representa la población. En el caso de PTPT dependerá de si se compensa la variabilidad presente. Recordemos de ML que en el caso de que el número de scores de entrenamiento tienda a infinito, ML converge al vector de parámetros óptimo: obviamente se reduce la variabilidad a 0. Para escoger el mejor modelo en el siguiente apartado se hará uso de las curvas ECE.

6.4. Medida de Rendimiento en los datos de entrenamiento

En este apartado se va a hacer uso de las curvas ECE para evaluar las hipótesis del apartado anterior. Además se va a utilizar la medida Cllr, (6), es decir, la ECE en O(Hp)=1 para realizar una comparativa del rendimiento total obtenido, mediante diagramas de barras.

6.4.1. Sobreentrenamiento gauss-KDF

A la vista da Figura 15 se puede concluir que efectivamente, como cabía esperar, gauss-KDF presenta una pérdida de información (ECE, curva roja) mucho más grande que ML, cosa que se observó que podría ser por sobreentrenar y por lo tanto por representar muy bien el conjunto de datos que modela, pero sin dejar suficiente libertad para modelar otro conjunto de datos. Esto se puede observar bien en las *s-LR* de A.3. Se pueden observar altas variaciones de LLR para un mismo *score*. Aunque a la vista parece que no, el eje y va de -100 a 800 valores de LLR por lo que no pueden apreciarse del todo estos grandes saltos. Se ha preferido dejar libertad a este eje para mostrar como de fuertes varían las propias curvas con *scores* en el eje x cercanos además de cómo de fuertes son los valores del LLR. Variaciones tan grandes se deben a variaciones bruscas en los modelos, fruto del ajuste a los propios datos.

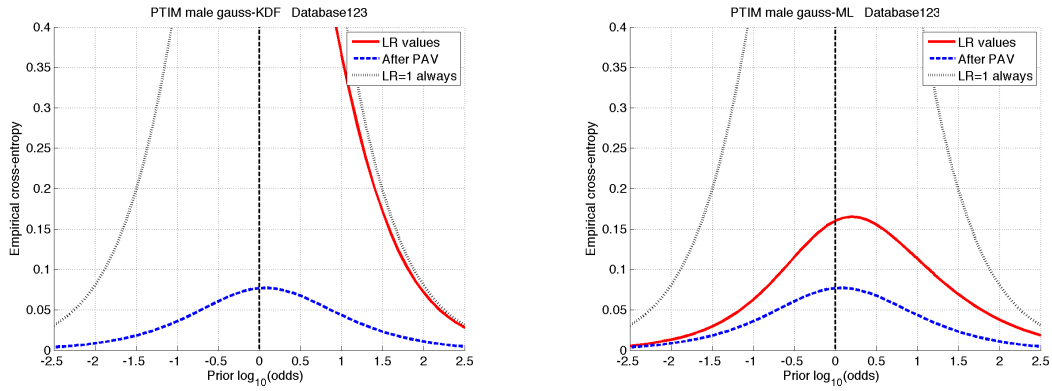


Figura 15: Comparativa de curva ECE para misma base de datos, genero y subdivisión (ver título figura). A la izquierda rendimiento para gauss-KDF, a la derecha rendimiento para gauss-ML.

Además en las pruebas realizadas la comparativa de costes computacionales entre algoritmos rondaba un ratio de unas 3500 veces más lento para KDF respecto a los otros dos. Esto es otro claro inconveniente. Por lo tanto gauss-KDF no es una técnica recomendable para cálculo de ratios de verosimilitud, lo que no implica que pueda ser interesante para otro tipo de aplicaciones.

Para terminar de mostrar visualmente a que nos referimos en este apartado se van a mostrar un par de histogramas.

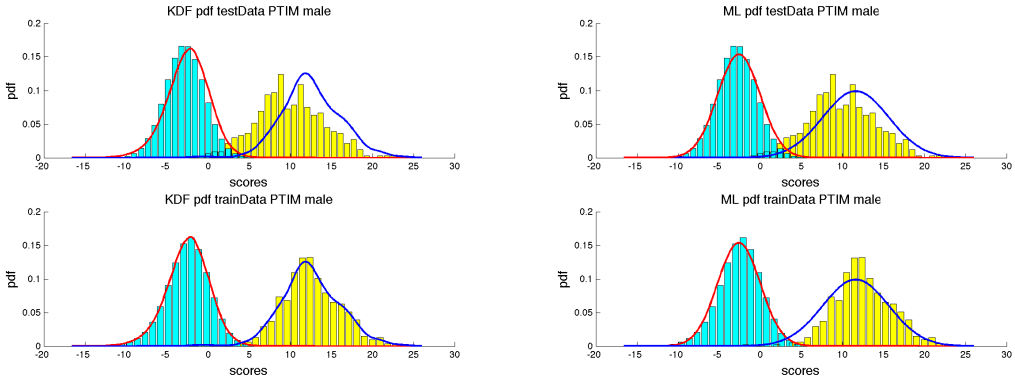


Figura 16: Histogramas para mostrar sobreentrenamiento de KDF. A la izquierda arriba: modelado con Database123 y datos Database4, abajo: Database 123 y datos Database123. A la derecha se representa el mismo efecto con ML. Se puede observar el sobreeajuste que produce KDF y como se manifiesta en los datos de de test, mediante una mala representación de los mismo (se pueden observar pequeñas variaciones en la fdp generada mediante KDF que ni mucho menos son representativas de los datos de test). ML sobreajusta menos a los datos de train con lo que abarca más datos de test.

6.4.2. Variabilidad en la población y outliers.

En este apartado se van a comprobar si finalmente PTPT tiene variabilidad en su población y se va a mostrar el efecto comentado de tener *outliers* en la subdivisión de validación.

Respecto a PTPT hemos observado como los modelos entrenados presentaban cierta variabilidad en la clase Hp. Por tanto es necesario comprobar si dicha variabilidad se ve compensada, es decir, si pese a que tenemos curvas que varían un poco siguen siendo representativas de la población. Esto indicaría que el conjunto de entrenamiento abarca ese rango de variación , o dominio en el que está definido la variable aleatoria que tratamos de modelar (ya que si se obtiene buen rendimiento para las 4 divisiones quiere decir que la división de *test* está siempre representada). Por lo tanto al entrenar el modelo con todos los datos se tendrá un modelo al menos estable para datos de la población de *train*.

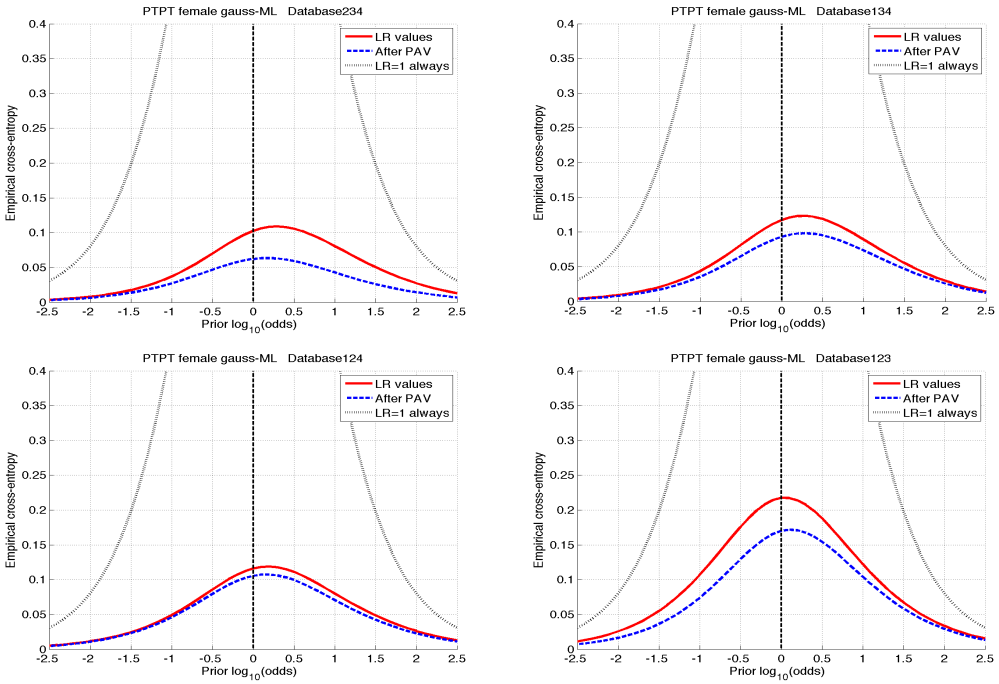


Figura 17: Curvas ECE para las cuatro posibles combinaciones de la validación cruzada (ver título de figuras) para mujeres PTPT y modelo ML. La baja variación de ECE es un indicativo de que la variabilidad queda reducida. Ello es un indicativo de que la población está bien representada por esos datos. Además viendo los modelos (ver A.1) la variabilidad no es excesiva. Puede deberese a una escasez de datos a la hora de subdividir las bases de datos.

Respecto a la aparición de *outliers*, viendo IMIM o PTIM podemos observar que uno de los modelos presenta variación respecto al resto (ver A.1). Dado que las curvas ECE (figura 18) solo presentan mal rendimiento para esta subdivisión. Ello puede ser un indicativo de que existen muchos *outliers* para entrenar, o en el conjunto al que se le aplica la *s-LR* en esa división de la validación cruzada, ya

que de no ser así se hubiera obtenido lo mismo que en la figura 17, es decir, atenuación de la variabilidad.

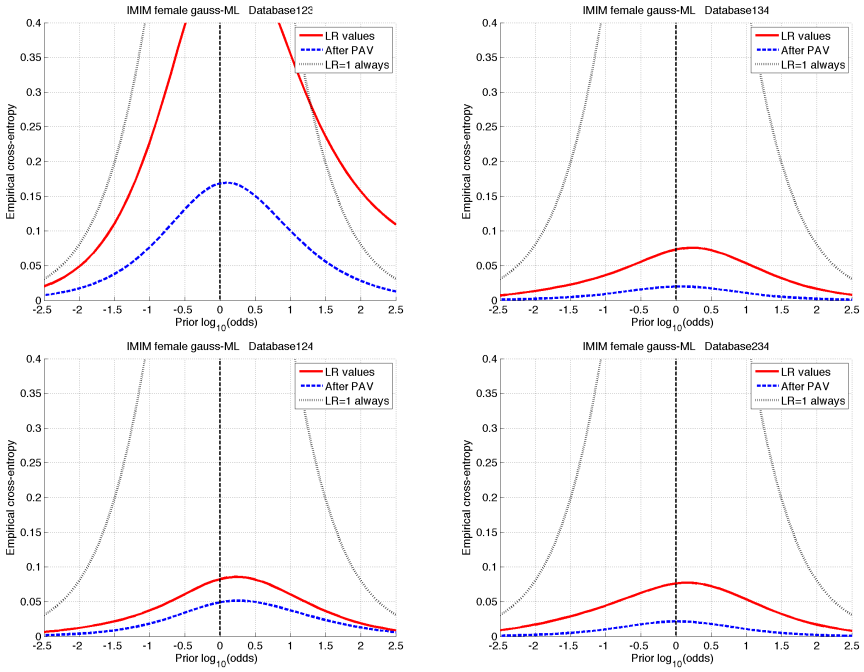


Figura 18: Curvas ECE para base de datos IMIM female entrenado mediante ML. Efecto producido por outliers.

Finalmente tanto RL como ML presentan resultados muy parecidos. Hay veces que ML consigue mejor resultado que RL y viceversa. Sin embargo RL nunca presenta un rendimiento peor del DP (curva azul) que ML por ser una función monótona creciente, aunque en la mayor parte de las veces ML no empeora el DP. Aun así el valor de la curva roja es el Cllr total por lo que mide el error en su conjunto. Dado que la curva roja siempre suele ser igual para RL como para ML no se puede destacar ninguna técnica sobre la otra en la mejora del error debido a la calibración.

Para finalizar se muestran diagramas de barras con el Cllr para todo el conjunto de pruebas realizadas mediante la validación cruzada. En ellas se pueden comprobar los efectos comentados en los apartados 6.4.2 y 6.4.1.

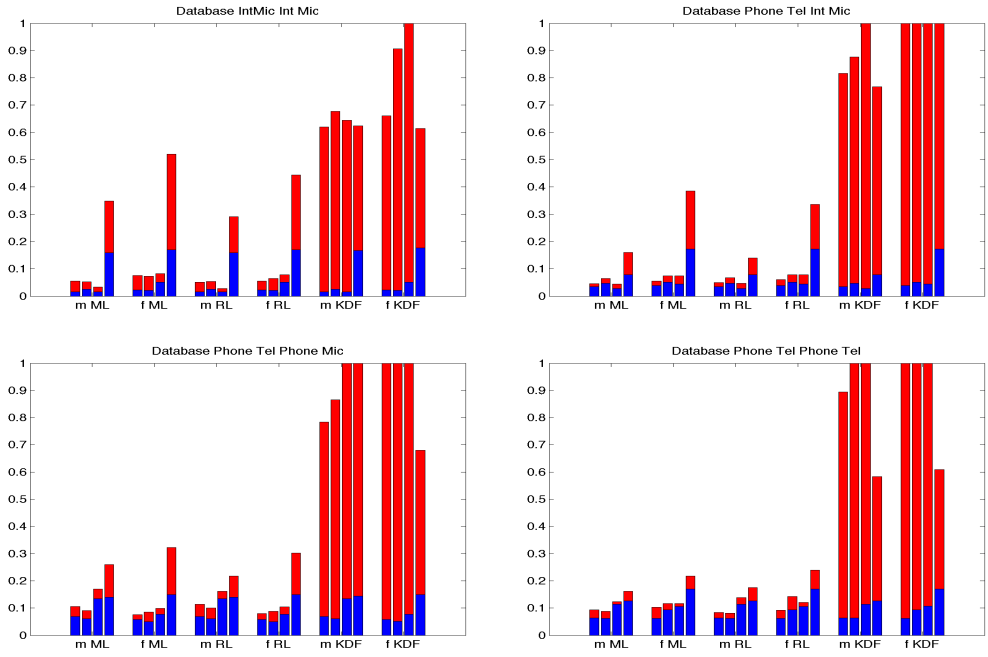


Figura 19: Se muestran diagramas de barras para cada una de las bases de datos (ver título). De izquierda a derecha Cllr para male ML, female ML male RL, female RL, male KDF y female KDF. La m hace referencia a male y la f a female, así m ML representa la base de datos de male entrenada con ML. Las 4 barras muestran las 4 k combinaciones de la validación cruzada. En azul se muestra el Cllr para DP y en rojo el Cllr por calibración.

6.5. Experimentos de validación o de *test*.

Una vez se han analizado los posibles efectos en la parte de entrenamiento el siguiente paso es probar los modelos con un conjunto de *test*.

Primero se presentará el protocolo experimental llevado a cabo.

6.5.1. Protocolo experimental

Para la realización de las pruebas de *test* se han utilizado dos modelados. Se obtienen los modelos a partir del conjunto entero de entrenamiento (modelo Total) y por otro lado se obtienen los modelos mediante la media de los parámetros obtenidos por cada una de las subdivisiones hechas en la validación cruzada (modelo mediaXval). Los resultados son muy parejos, y por lo tanto usaremos el modelo Total o media Xval en los siguientes experimentos, indistintamente.

Además se han realizado dos tipos de pruebas denominadas calibración *pool* y calibración por condición, siempre separando los hombres de las mujeres:

- **Calibración Pool (CP):** Consiste en juntar todos los datos de entrenamiento de todas las bases de datos, siempre separando entren hombres y mujeres, entrenar un modelo y juntar todos los datos de test para realizar el cálculo de LR.

- **Calibración por condición (CD):** En este caso para cada base de datos (es decir, para PTPT, PTPM, IMIM y PTIM) se generan unos modelos y se calculan s-LR para cada base de datos. Se generan LR con un conjunto de test generando el LR para cada condición con la *s-LR* correspondiente.

Además, se va a presentar curvas ECE de los LR generados para la calibración por condición para cada base de datos debido a que se observó que se obtenían mejores rendimientos para CP que para la CD, algo que por lo que se comentará más adelante no era el resultado esperado.

6.5.2. Curvas ECE: calibración *pool* y calibración por condición

A continuación se muestran las curvas ECE para las dos pruebas realizadas. Debido a la limitación de espacio para la realización de la memoria y a que lo que se pretende mostrar en este apartado es la diferencia entre *pool* y condición y entre ML y RL, no se presentarán todas las gráficas.

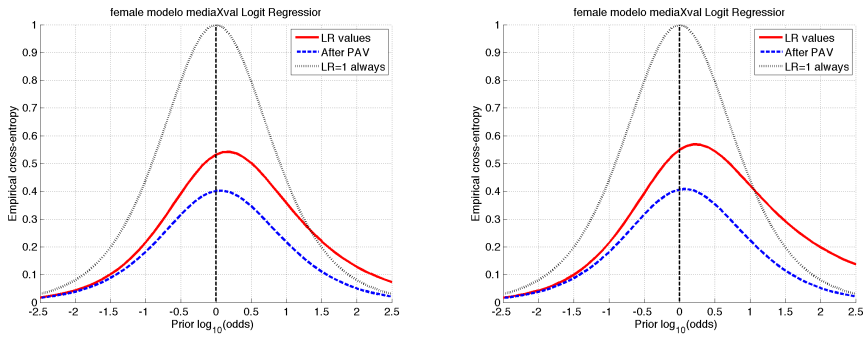


Figura 20: Curva ECE prueba de test y modelo RL. A la izquierda calibración pool, a la derecha caliración por condición.

En la figura 20 se puede observar como CP presenta mejor rendimiento que CD, este comportamiento se daba en todas las gráficas obtenidas para todos los modelos y géneros. A priori parece ilógico que un modelo generado para todos los *scores* mezclados arroje mejor rendimiento que uno entrenado para cada condición de grabación. Ello motivo el análisis realizado en la siguiente sección.

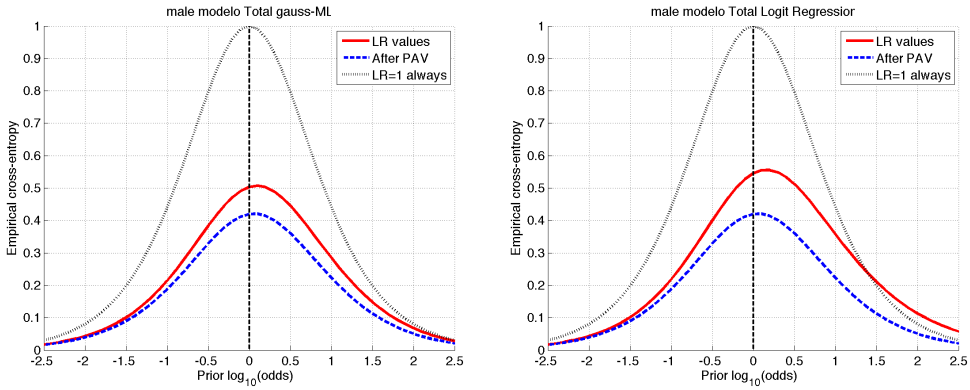


Figura 21: Curva ECE comparativa ML vs RL, y esquema CP.

Además el modelado con gauss-ML arroja mejores resultados que con RL (figura 21). Ello es debido a que si las distribuciones que arroja el nivel de discriminación siguen pdf gaussianas, parece lógico que transformaciones *s-LR* de carácter cuadrático (fruto de ratios de verosimilitud a partir de pdf gaussianas) apoyen mejor las decisiones que transformaciones *s-LR* lineales.

6.5.3. Análisis de Int Mic Int Mic

En el apartado anterior se vió como CD arrojaba peores resultados que CP. Ello motivo el análisis individualmente del rendimiento de cada una de las bases de datos con el modelo entrenado para la misma. Así se podía ver que *s-LR* estaba siendo el motivo de que CD fuese peor que CP.

Por sorpresa, el resultado encontrado fue el que se describe a contiuación (ver figura 22). Esto motivó el análisis realizado en esta sección y permitió establecer un conjunto de hipótesis acerca de las distintas bases de datos y la relación entre los *scores* de *train* y los de *test*.

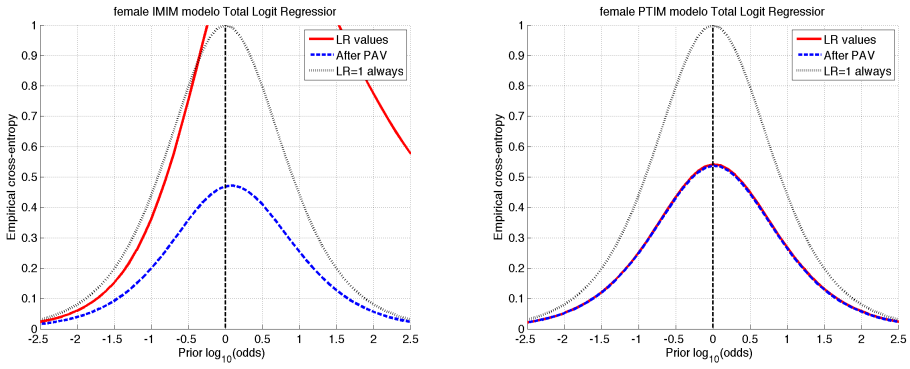


Figura 22: Curva ECE para LR generados a partir de modelos y conjuntos de datos de test provenientes de la misma población (indicada en el título).

Se observa en la Figura 22 que en IMIM el modelo tiene problemas de desajustes entre los datos de *train* y *test*. De ahí que con la CD se obtengan mejores resultados pues de alguna manera se reduce este efecto de desajuste al haber influencia de todas las bases de datos en el modelo. Se puede afirmar que está ocurriendo este tipo de desajuste porque en la parte de entrenamiento no se observaron problemas de los citados para IMIM: el conjunto de *scores* de *train* representaba adecuadamente dicho conjunto (no variabilidad de los datos, observado en adecuados rendimientos de la validación cruzada y en modelos con poca variabilidad, ver A.1). Además respecto a la influencia de los *outliers* se puede observar en la figura 22 derecha como los *outliers*, al ser pocos, no interfieren apenas en el rendimiento (recordar como en el conjunto de entrenamiento PTIM presentaba un indicio de contener *outliers*). Además la variabilidad presente en PTPT, tal y como se comentó, queda compensada (figura 23).

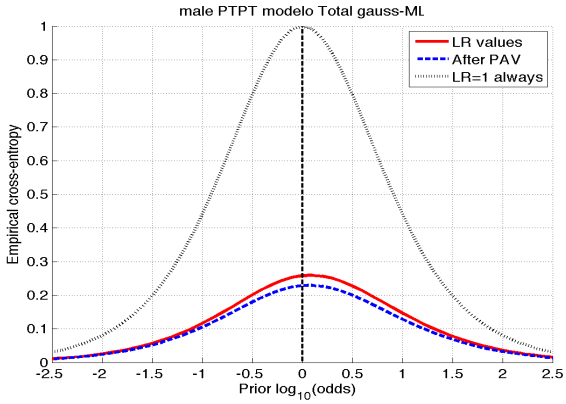


Figura 23: Curva ECE para base de datos PTPT. Se observa un rendimiento adecuado.

Para analizar en mayor profundida el problema observado se van a presentar los histogramas de los datos de *test* junto a los modelos (figura 24). Además en base a ello se van a exponer una serie de conclusiones interesantes:

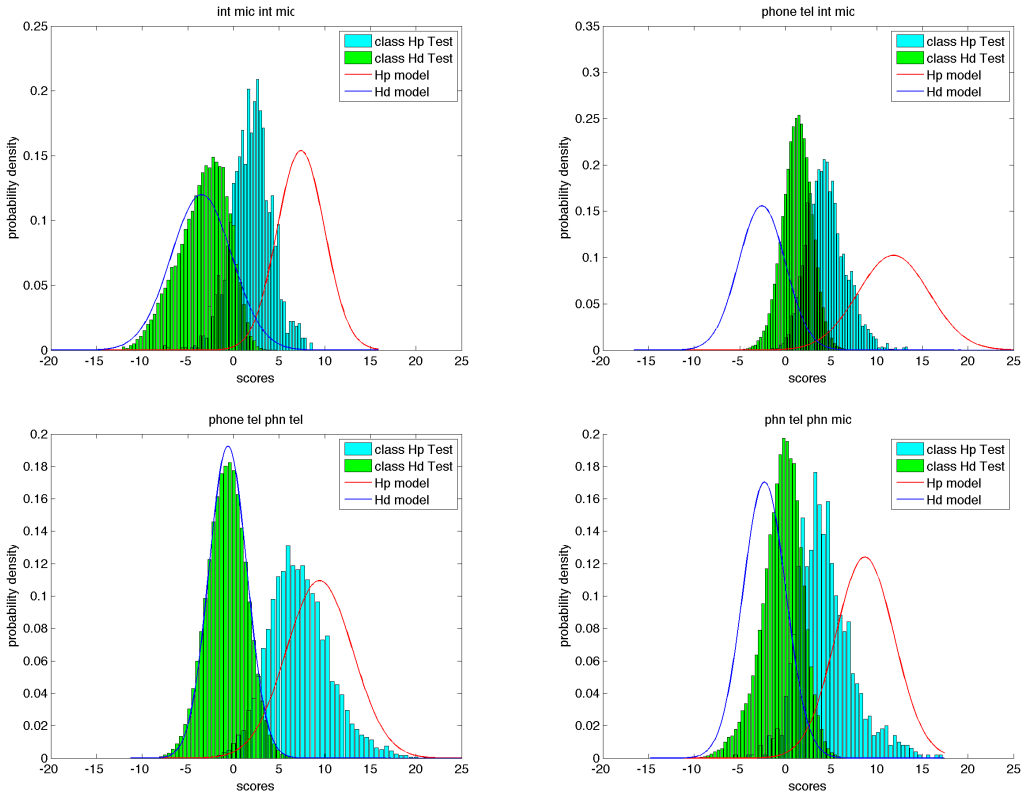


Figura 24: Histogramas con los datos de test junto a los modelos generados en la parte de train para la CD.

A la vista de la figura 24 se puede observar como todas las bases de datos presentan desajustes entre el conjunto de *train* y el de *test*. Sin embargo éste no puede ser el único motivo de que IMIM presente tan mal rendimiento, pues habría mal rendimiento para el resto de bases de datos también. Dicho esto el objetivo de este apartado es hacer un pequeño análisis de cómo influye el DP.

Observando el DP presente en las bases de datos (figura 25), se puede observar como las bases de datos de *test* presentan peor DP que las de *train*. Por lo tanto la primera conclusión interesante es que es preferible que los *scores* de *train* presenten peor DP y los de *test* mejor ya que ello se verá reflejado en modelos más juntos que los datos que deben modelar. Viendo la figura 24 equivaldría a que los datos del histograma azul estuviesen a la derecha del modelo, en vez de a la izquierda y lo mismo para los datos del histograma verde, en este caso a la izquierda del modelo. El motivo es que así los datos de cada clase siempre presentarían una verosimilitud mayor para el modelo de su clase que para el modelo de la clase contraria, reduciendose así el solape entre histogramas.

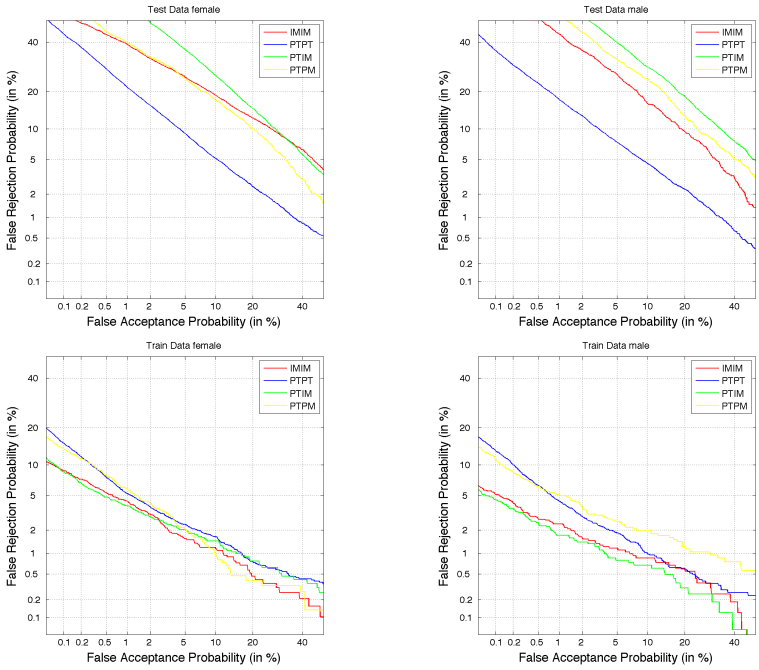


Figura 25: Curvas DET para hombres (derecha) y mujeres (izquierda). Arriba tenemos el conjunto de test y abajo el conjunto de train. Efectivamente el conjunto de train presenta mejor DP que el conjunto de test.

Dentro del desajuste presente, es interesante destacar que es mayor para la clase *target* debido a que para esta clase el número de datos disponibles es menor lo que como ya se ha visto puede acrecentar el efecto de una mayor variabilidad de los datos. De hecho es interesante destacar que por las condiciones de grabación de IMIM, mucho más variables que el resto por las condiciones de entrevista y por el uso de múltiples micrófonos, aparecen subpoblaciones que originan una variabilidad todavía mayor, reflejada en un gran desajuste *train-test* en

comparación al resto de datos. Por lo tanto IMIM es una base de datos de la que se requieren más datos de los disponibles para poder compensar la gran variabilidad que presenta. Aun así este desajuste se produce en el resto de bases de datos lo que permite concluir lo siguiente, ya que por ejemplo PTIM es la que mayor desajuste presenta:

- En el caso de que haya desajuste, es preferible que éste se produzca más o menos en la misma medida para ambas clases (como en PTIM): es interesante que la distancia de los datos de una clase a la media de la distribución que modela dicha clases (se habla de la media por simplificar el ejemplo pero la varianza también influye) sea menor o igual a la distancia a la clase contraria, como ocurre con PTIM y se ve reflejado en su rendimiento (figura 22). Si esto no ocurre (como en IMIM) a la vista de la figura 24 la clase target contiene muchos de esos datos fuera del modelo Hp y en gran parte de ellos la función de verosimilitud es mayor para la clase Hd que para Hp, lo que provoca LRs que apoya con gran fuerza la proposición incorrecta viéndose reflejado en un aumento brusco de ECE y Cllr.

- Un bajo DP hará que los modelos para ambas clases estén juntos por lo que los datos no representarán la población de una manera discriminatoria, haciendo que cualquier pequeño outlier (fruto del desajuste *train-test*) obtenga verosimilitudes mayores para la clase contraria. En el extremo opuesto con un DP muy alto se produce desajuste como ya se ha comentado en el párrafo anterior. Para un bajo DP reducir la variabilidad al máximo es importante por lo que se acaba de comentar: si se tienen modelos variables y además están cerca entre ellos, el efecto de los *outliers* puede ser importante entre entrenar con un conjuntos de datos (que den lugar a un modelo) y otro conjunto de datos (que den lugar a otro modelo distinto) viendose reflejado en una influencia en el mínimo error de bayes, (Duda, Hart, & Stork, Pattern Classification, 2000), ver figura 26.

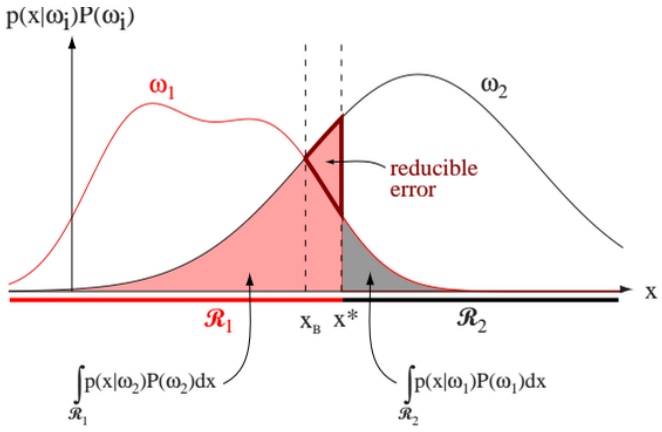


Figura 26: En la figura se muestra el mínimo error de Bayes según la teoría de decisión bayesiana para las probabilidades a posteriori de dos clases. Cuanto más juntas estén las funciones de verosimilitud, mayor es el error de la decisión usando el umbral de Bayes. En esta imagen w es H y x es s.

Finalmente un efecto interesante a observar es, en las transformaciones *s-LR* (figura A.1), observar como la *s-LR* apoya decisiones *target* para datos que como se puede observar sobre el eje x están más a la izquierda que otros datos a los que se apoya la decisión *non-target (*ver los puntos de la *s-LR* en donde cambia la curvatura*)*. Básicamente si se tiene un *score* con un valor de -10 (ver A.1 cualquier *s-LR* que no sea la de IMIM) se puede observar como debido al cambio de curvatura un *score* con valor de -15 obtendrá un mayor apoyo para la clase *target* mientras que -10 obtiene mayor apoyo para la *non-target* algo que si la *s-LR* es coherente, por motivos obvios, no debe de llevar a este tipo de incongruencias. Esto lleva a concluir que ML funciona bien en aquellas zonas representadas por los datos (*scores* de entrenamiento). Fuera de estas zonas (ya que las gaussianas tienen dominio infinito, por lo que el dominio de la *s-LR* es todo $\Re$) el valor de la *s-LR* no es representativo por lo que dicha transformación puede comportarse de manera extraña (hay que tener en cuenta que las pdf integran a 1 por lo tanto en función de la desviación típica que presenten las gaussianas, fuera de los valores representativos de las mismas, los valores de la pdf pueden ser mayores para gaussianas cuya desviación típica es mayor. Por tanto puede ocurrir que un *score* que cae en una zona no representativa, por ejemplo muy a la izquierda de la clase *non-target,* presente una función de verosimilitud mayor para la clase *target,* pese a que está más cerca de la media de la clase *non-target* que de la media de la clase *target*). En el caso de que la desviación típica de Hp sea mayor que la de Hd ocurre este efecto de apoyar como *target* datos *non-target*. En el caso contrario el efecto es al contrario (observar la curvatura del *s-LR* para IMIM A.1). Se puede observar como la *s-LR* en este caso es convexa en lugar de cóncava.

Este efecto también es minimizado por la estimación bayesiana cuando hay pocos datos, debido a que las colas de las t-student se levantan.

6.6. Simulación de Casos Forenses Reales

En este último apartado se van a simular casos forenses reales, donde existirá escasez de *scores target* de entrenamiento fruto del esquema de *anchoring* propuesto. El objetivo es mostrar que la estimación bayesiana funciona mejor que la estimación de máxima verosimilitud en dicho escenario de escasez de datos.

Recordando de la sección de *anchoring*, este concepto surge como consecuencia de fijar ciertas características de las hipótesis del caso forense para adaptar los *scores* de entrenamiento a dichas clases. Eso suele implicar que los *scores target* de entrenamiento deben estar generados por el sospechoso en cuestión. En este ejemplo se van a proponer dos esquemas de *anchoring* diferentes con las siguientes hipótesis:

- Primer esquema:
    - Hp: La voz dubitada e indubitada pertenencen a la misma persona y la voz indubitada la generó el sospecho en unas condiciones de grabación determinadas.
    - Hd: La voz dubitada e indubitada no pertenecen a la misma persona y la voz indubitada la generó el sospechoso en unas condiciones de

grabación determinadas. Además la voz dubitada fue generada por un locutor perteneciente a una población de potenciales autores.

- Segundo esquema. Este esquema tiene las mismas hipótesis que el anterior. Sin embargo, para ajustar mucho más las condiciones de los *scores* de entrenamiento al caso, se añade esta particularidad:

    - La voz indubitada es conocida y por tanto las locuciones utilizadas para generar las puntuaciones con las que entrenar los modelos deberán contener esa misma locución indubitada.

El procedimiento a seguir es el siguiente: para un caso forense simulado, un *score* de la base de datos simula el obtenido a partir de la comparación de las locuciones dubitada e indubitada que provendrían del juzgado (las pruebas). Los datos de entrenamiento para ese caso concreto se obtienen de la siguiente manera:

- En el primer esquema, los *scores target* de entrenamiento son todos los *scores target* que no contienen la locución indubitada del caso. Los *scores non-target* de entrenamiento son *scores non-target* obtenidos con voces del sospechoso que no son la indubitada del caso, y locuciones de otros locutores.

- En el segundo esquema, los scores *target* de entrenamiento son todos los *scores target* que contienen la locución indubitada del caso, salvo el score del caso si fuera *target*. Los scores *non-target* de entrenamiento son scores *non-target* obtenidos con la voz indubitada del caso y locuciones de otros locutores, salvo el *score* del caso si fuera *non-target*.

Para la simulación de la prueba como *scores non-target* el procedimiento es similar al comentado en el párrafo anterior. Por supuesto solo se consideran los datos que cumplen con el anclaje expuesto, que dependerá de la base de datos. Hay locutores de los que se disponen más *scores non-target* que de otros. De todas maneras el número de *scores non-target* es parecido y suficiente como para modelar correctamente la clase *non-target*.

Este procedimiento de simulación de casos se repetirá para cada *score* de la base de datos, dando lugar a un número de casos simulados igual a número de *scores* en la base de datos, salvo para la clase *non-target* en la que se simularán 10 casos diferentes por cada modelo.

Así por ejemplo imaginemos que simulamos un caso con el segundo esquema en el que la prueba proviene de una voz indubitada M1F1 (es decir, locución 1 (F1) del locutor 1 (M1)) y una voz dubitada T250 (es decir, locución de test 250). En este caso, aunque en el caso no lo sabríamos, en la simulación se sabe que dicha comparación da lugar a un score *target*. Por lo tanto, se escogerán como datos de *train* todos los scores que se generen con M1F1 y los ficheros de test diferentes de T250, ya que no puede formar parte del modelo un dato que no se sabe si es *target* o *non-target*. Para el primer esquema si el dato proviene de M1F1 T450 y da lugar a un score *target*, se escogerán como datos *target* de *train* todos los scores *target* que generen las locuciones de M1

diferentes de F1, y los ficheros de test diferentes de 450. Para las simulaciones en las que la prueba es un dato *non-target* (ya que hay que comprobar el rendimiento obtenido para casos en los que el sospechoso es culpable y casos en los que no) los datos de *train non-target* utilizados serían para el primer esquema todos los M1 con locuciones diferentes de F1 y test distintos a 450. Para el segundo esquema se escogerán datos *non-target* generados con un test distinto de 250 pero con la locución siempre F1.

Así pues los resultados se muestran en la figura 27 donde queda de manifiesto la ventaja que la estimación bayesiana aporta en condiciones de pocos datos y por lo tanto la ventaja que aporta a la hora de establecer ciertos esquemas de *anchoring*. Los hiperparámetros escogidos son los mismos que permitían hacer la probabilidad a priori del parámetro no informativa (ver figura 6).

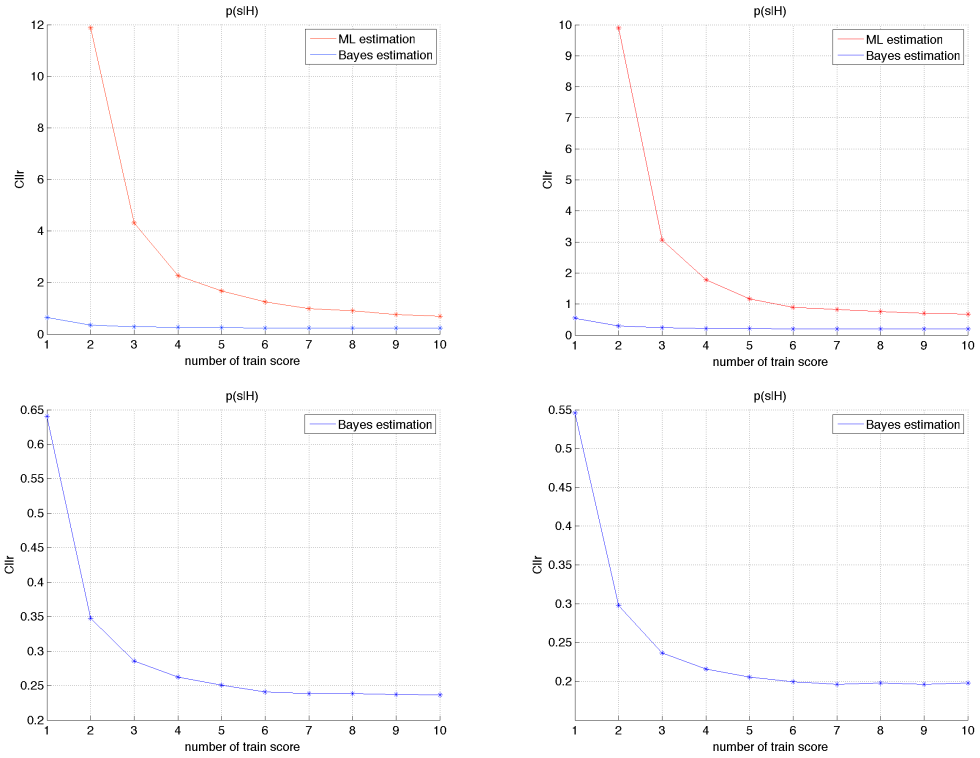


Figura 27: Cllr para el experimento de simulación de casos reales. A la izquierda se muestra el primer esquema y a la derecha el segundo. Las figuras de abajo representan el Cllr solo para estimación bayesiana, es decir, son ampliaciones del eje y de las figuras de arriba.

Además de obtenerse mejor rendimiento usando estimación bayesiana que máxima verosimilitud se puede observar como el segundo esquema de anclaje es mejor que el primero. Esto es debido a que aunque el reconocimiento de locutor que se usa en este trabajo es independiente de texto, está claro que el sistema no es totalmente invariante al texto a la hora de arrojar los *scores* correspondientes. Además, si queremos simular las condiciones del caso a la hora de utilizar los *scores* de entrenamiento, parece razonable que el habla indubitada sea lo más parecida en el *scores* del caso que en los scores de entrenamiento. Los resultados corroboran dicha hipótesis.

Los resultados obtenidos se han calculado a partir de un número de LR *target*: 18192 y un número de LR *non-target:* 11560, como mínimo. Este es el número de LR utilizados para calcular Cllr con 10 datos de entrenamiento. Debido a que había modelos para los que el número de datos de entrenamiento era menor a 10, para 10 solo se han utilizado aquellos que tenían al menos un conjunto de *scores target* de entrenamiento igual a esta cantidad. Así mismo se ha procedido para el resto, de manera que las medidas de Cllr para cada número de datos de entrenamiento están calculadas en las mismas condiciones, tanto para ML y MB como para el primer y segundo esquema.

7. Conclusiones y Trabajo Futuro

En este TFG hemos simulado la problemática forense de interpretación de evidencias utilizando sistemas automáticos de reconocimiento de locutor y bases de datos desafiantes provenientes de evaluaciones NIST. Las principales conclusiones y aportaciones del TFG son las siguientes:

- En este trabajo se ha comprobado que efectivamente la estimación bayesiana arroja mejores resultados que la estimación ML en entornos simulados reales, en los que la escasez de datos de entrenamiento es un factor relevante

- Por otro lado se han comparado dos esquemas de *anchoring*. Dichos esquemas se pueden denominar de anclaje a la locución o de anclaje al locutor. Son dos propuestas originales de modelado de entornos reales forenses primero porque está usado en un entorno real de *anchoring* y segundo porque es una comparativa hasta el día de hoy no hecha.

- Además se ha realizado un estudio de los problemas típicos que se pueden encontrar en cálculo de LR para interpretación en ciencia forense, comparando varios algoritmos clásicos para generar transformaciones *s-LR* y comentando la problemática existente así como diferentes propuestas, hipótesis, o requerimientos interesantes en el conjunto de datos para obtener mejores rendimientos. Además se ha visto como dicha problemática puede influir a la hora de generar *s-LR* en lo denominado calibración *pool* y calibración por condición, viendo en función de las condiciones cual funciona mejor.

- Finalmente, se ha probado a generar modelos mediante la media de los parámetros obtenidos en la validación cruzada junto a modelos generados con todos los datos de entrenamiento. No se han obtenido resultados interesantes ya que eran muy parecidos.

- Este trabajo abre varias vías futuras de trabajo entre las que se incluyen:

    - Usar modelos más complejos que el gaussiano para ser entrenados mediante MB

    - Ampliar las disciplinas forenses, tanto biométricas (huella, cara…) como de otro dipo (vidrios, marcas de pintura, etc).

8. Referencias

B. Hepler, A., P. Saunders, C., J. Davis, L., & Buscaglia, J. (January de 2012). Score-based likelihood ratios for handwriting evidence. *Forensic Science International* , 129-140.

Brümmer, N. (May de 2011). Fully Bayesian Score Calibration assuming Gaussian Distributions. *Agnitio Labs, South África* .

Swart, A., & Brümmer, N. (June de 2014). Bayesian calibration for forensic evidence reporting. *AGNITIO Research, South África* .

Ramos, D., & González-Rodriguez, J. (May de 2013). Reliable support: Measuring calibration of likelihood ratios. *Forensic Science International* , 156-169.

Minka, T. (2001). Inferring a Gaussian distribution.

D. van Leeuwen, D., & Brümmer, N. (2007). *An Introduction to Application-Independent Evaluation of Speaker Recognition System*. Springer.

Doddington, G. (1998). Sheep, Goats, Lambs and Wolves A Statistical Analysis of Speaker Performance in the NIST 1998 Speaker Recognition Evaluation.

Navrátil, J., & Ramaswamy, G. The awe and mystery of t-norm. *IBM T.J. Watson Research Center, Yorktown Heights*. NY.

Martyna, A., Zadora, G., Ramos, D., & Aitken, C. (January de 2014). Statistical Analysis in Forensic Science: Evidential Values of Multivariate Physicochemical Data. .

Zadora, G., Ramos, D., Martyna, A., & Aitken, C. (2014). *Statistical Analysis in Forensic Science: Evidential Values of Multivariate Physicochemical Data. John Wiley and Sons*. Wiley.

Ramos , D. (November de 2007). Forensic Evaluation of the Evidence Using Automatic Speaker Recognition Systems.

Ramos, D., Fiérrez, J., Krish, R. , & Meuwly , D. (2015). From Biometric Scores to Forensic Likelihood Ratios. En *Springer*.

Ramos, D., Gonzalez-Rodríguez, J., Zadora, G., & Aitken, C. (September de 2012). Information-Theoretical Assessment of the Performance of Likelihood Ratio Computation Methods. *Journal of Forensic Sciences* .

Duda, R., Hart, P., & Stork, D. (2000). *Pattern Classification*. Wiley.

Peebles, P. (2006). *Probability, Random Variables and Random Signal Principles*. 3.

Stafylakis, T., Kenny, P., Ouellet, P., Perez, J., Kockmann, M., & Dumouchel, P. (2013). I-Vector/PLDA Variants for Text-Dependent Speaker Recognition. *CRIM*. Montreal.

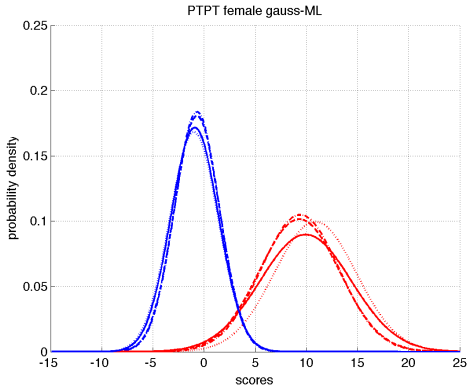
Lozano-Diez, A., Gomez-Piris, I., Franco-Pedroso, J., Gonzalez-Dominguez, J., & Gonzalez-Rodriguez, J. (2014). Speaker Clustering for Variability Subspace Estimation. *Iberspeech*.

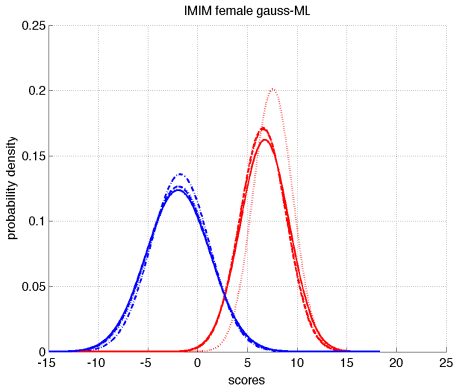
Gonzalez-Rodríguez, J., Ramos, D., Zadora, G., & Aitken, C. (September de 2012). Information-Theoretical Assessment of the Performance of Likelihood Ratio Computation Methods. *Journal of Forensic Sciences* .

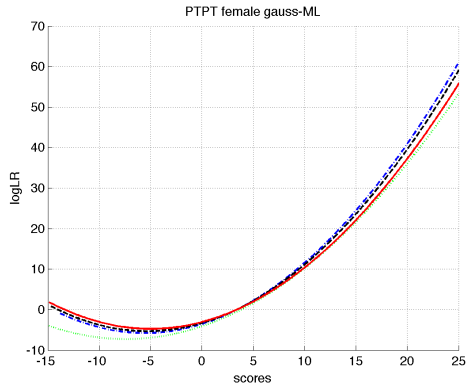
Anexo A: Modelos y Ratios de Verosimilitud

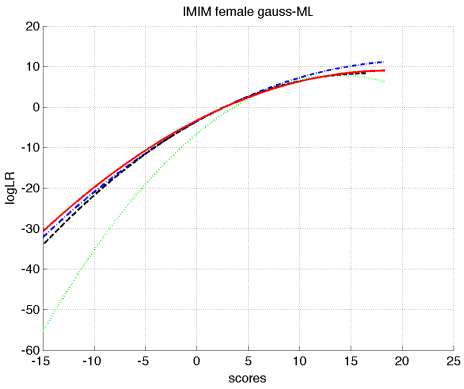
En el siguiente anexo se muestran los modelos y las transformaciones *s-LR* obtenidas en el proceso de validación cruzada para Regresión Logística, Máxima Verosimilitud Gaussiana y *Kernel Density Functions* Gaussianas.

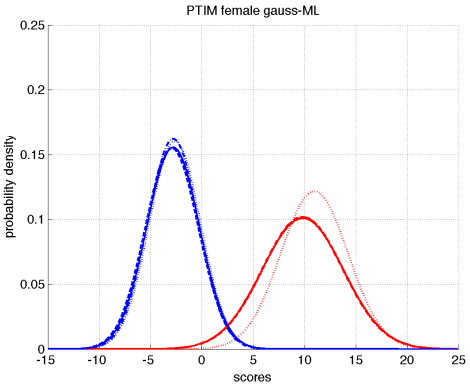
A.1 ML Gaussiano

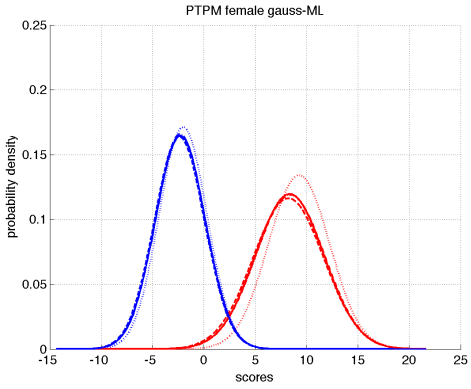












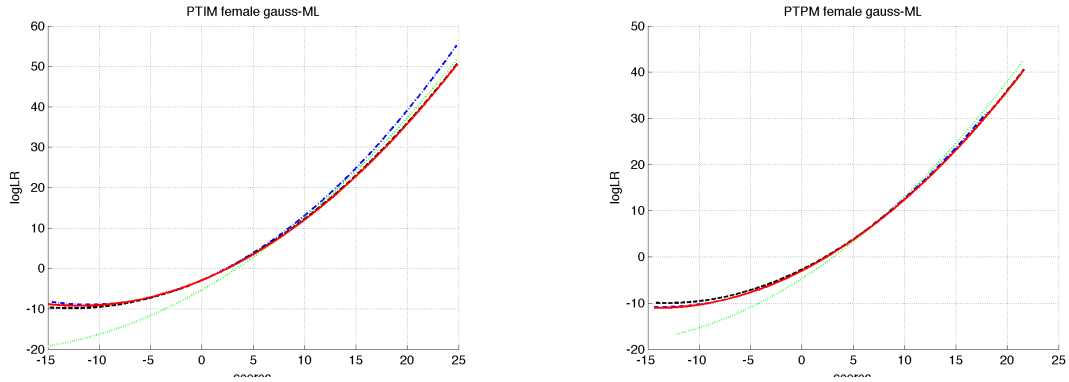
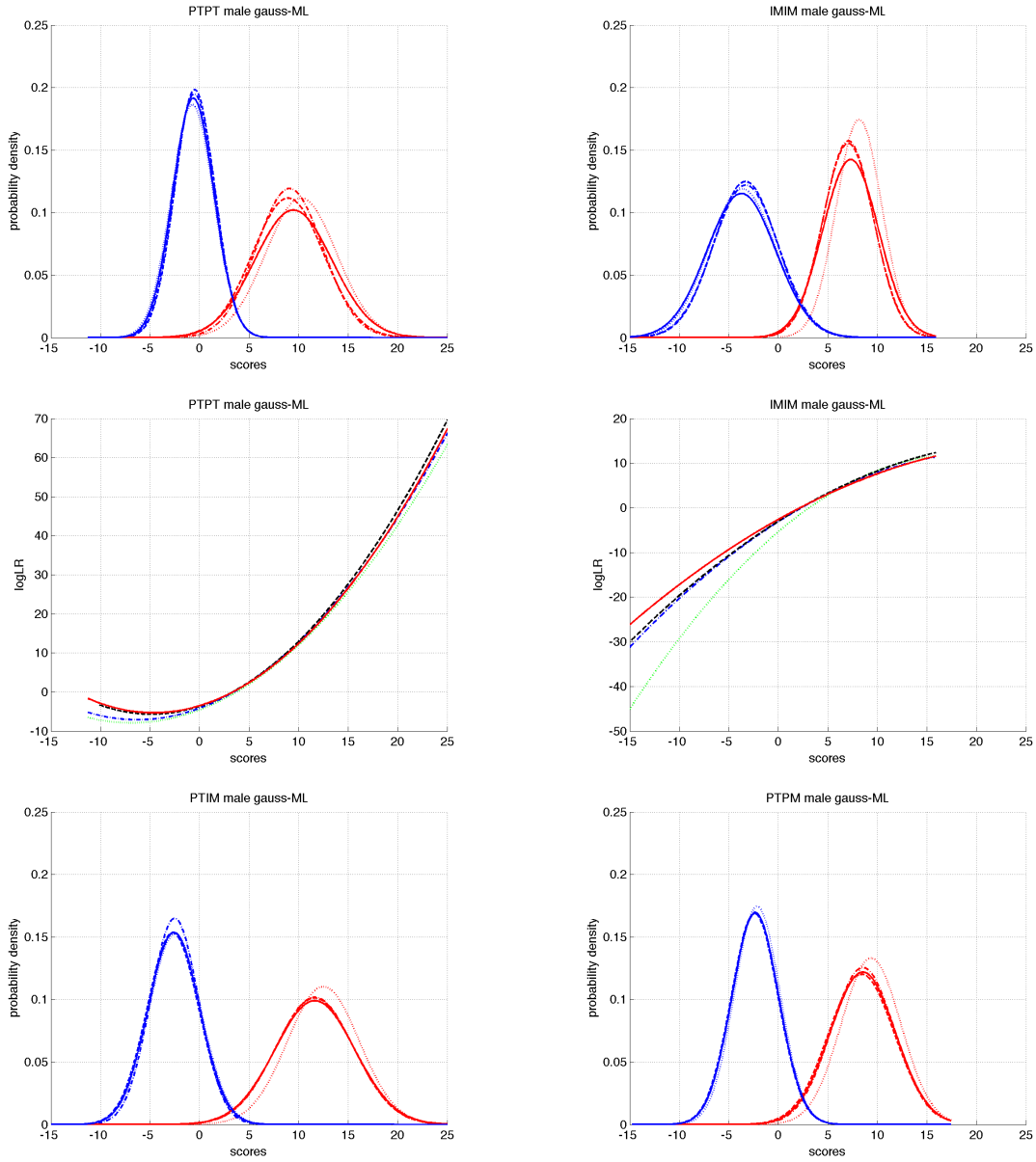


Figura A.1.a: ML gaussiano para base de datos de mujeres



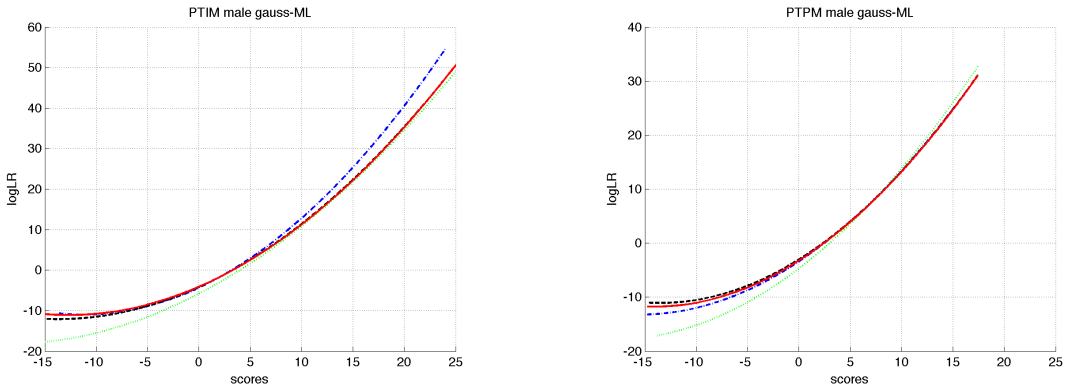


Figura A.1.b: ML gaussiano para base de datos de hombres.

Figura A.1: Se muestra el resultado de la generación de modelos y sus correspondientes transformaciones s-LR para ML gaussiano. En los modelos el de la derecha (rojo) representa la clase target y el de la izquierda (azul) la non target.

A.2 Regresión Logística

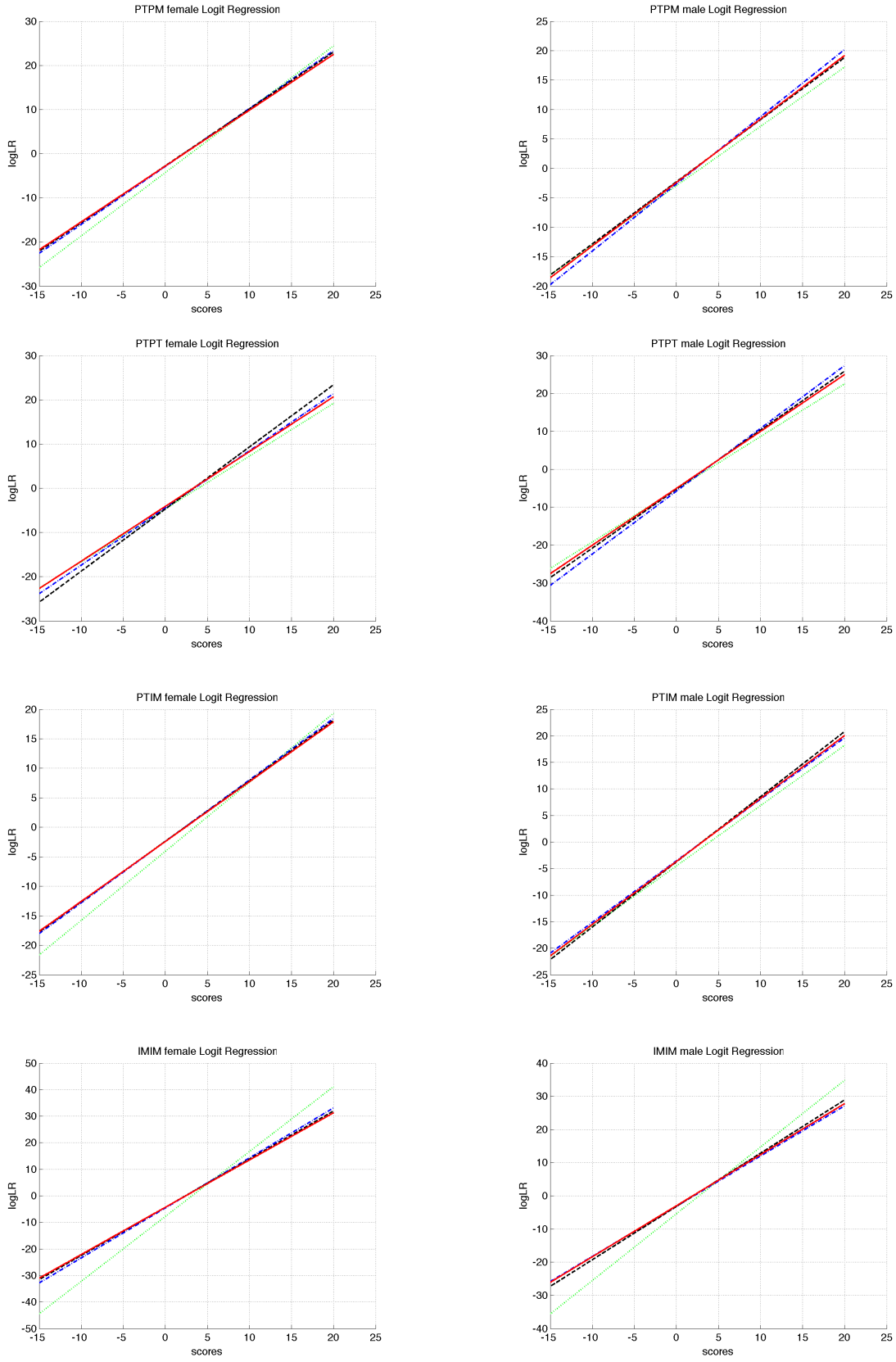


Figura A.2: Transformación score-LR para Regresión Logística. A la izquierda mujeres y a la derecha hombres.

A.3 KDF Gaussiano

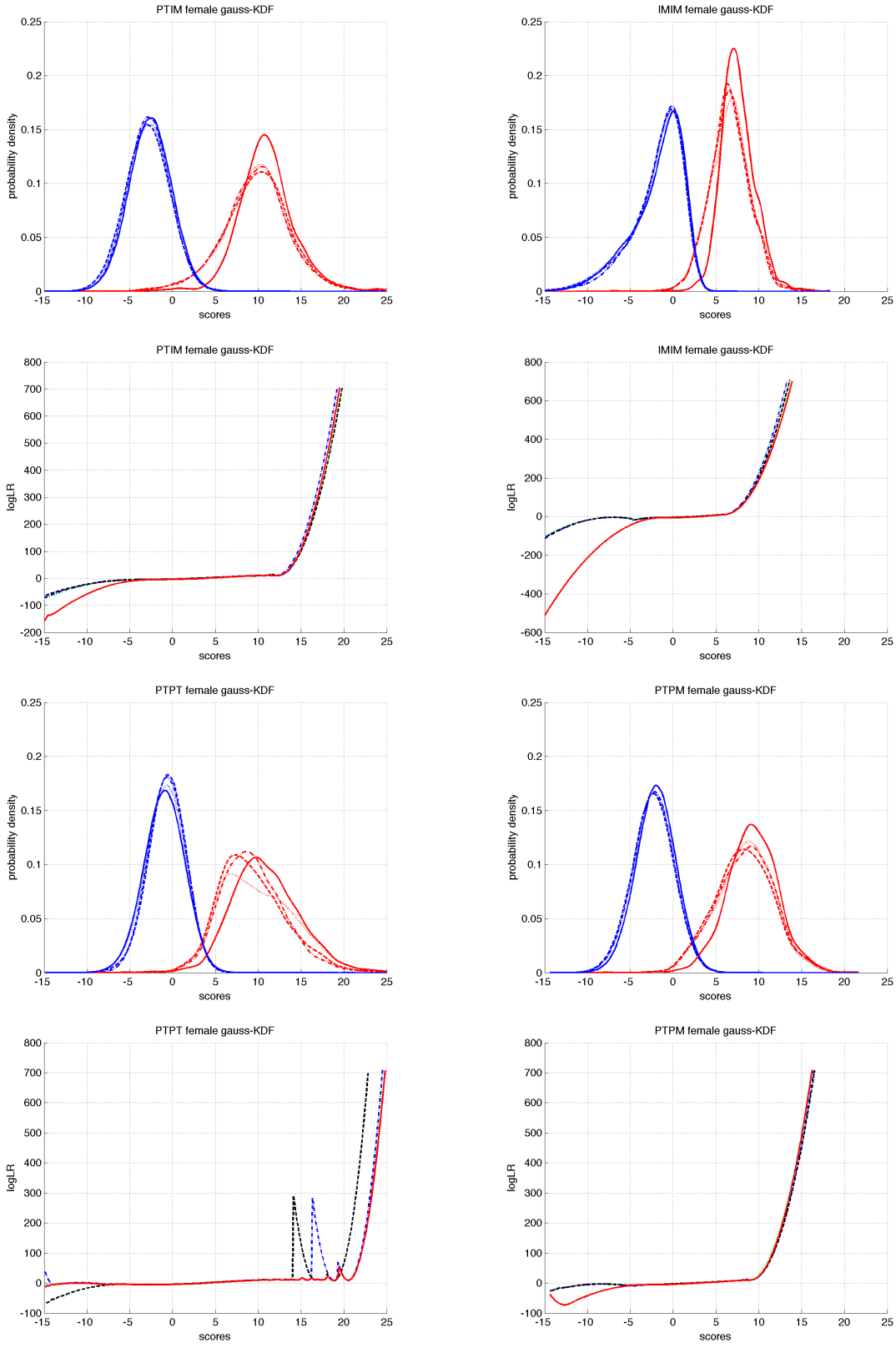


Figura A.3.a: KDF gaussiano para base de datos de mujeres.

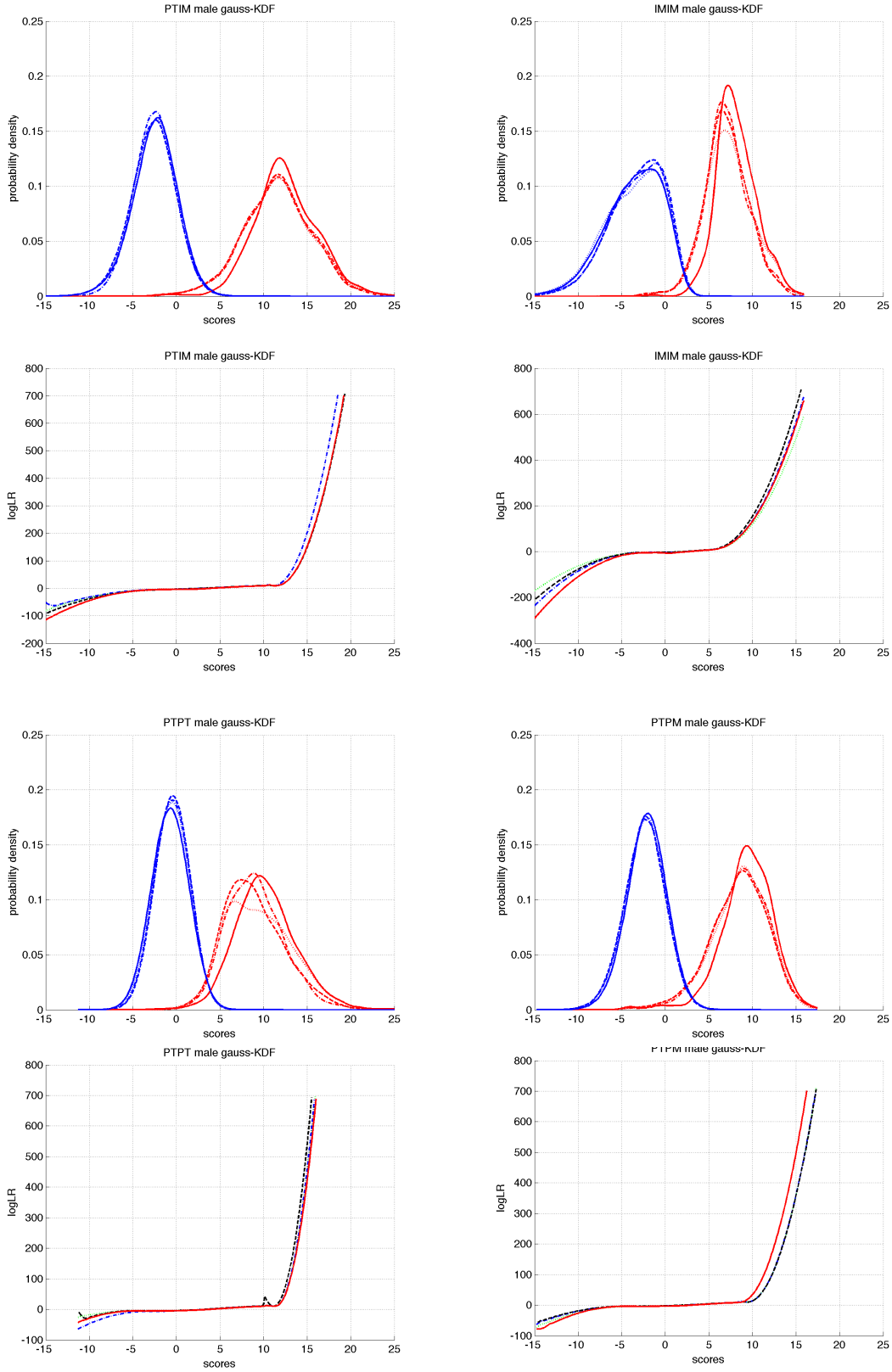


Figura A.3.b KDF gaussiano para base de datos de hombres.

Figura A.3: Se muestra el resultado de la generación de modelos y sus correspondientes transformaciones s-LR para KDF gaussiano. En los modelos el de la derecha (rojo) representa la clase target y el de la izquierda (azul) la non target.